

**IDENTIFICACIÓN Y EVASIÓN DE OBSTÁCULOS  
MEDIANTE FUSIÓN SENSORIAL PARA ROBOTS MÓVILES**

**JORGE IVÁN CÓRDOBA SÁNCHEZ**

**DIRECTORA**

**Ing. SANDRA NOPE, Ph.D.**

**DIRECTOR**

**Ing. HUMBERTO LOAIZA, Ph.D.**

**UNIVERSIDAD DEL VALLE  
ESCUELA DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA  
INGENIERÍA ELECTRÓNICA  
SANTIAGO DE CALI**

**2015**

Nota de Aceptación

---

---

---

---

---

---

Directora  
Sandra Nope Ph.D.

---

Director  
Humberto Loaiza Ph.D.

---

Jurado

---

Jurado

## RESUMEN

En este documento se presenta la fusión sensorial usando dos sensores: una cámara monocular que entrega imágenes en RGB y un sensor láser que proporciona medidas de distancia en algunos puntos. El resultado de la fusión se usa para que un robot móvil pueda realizar navegación tipo **wandering** en un ambiente de prueba estructurado.

Los algoritmos de navegación en general sólo detectan y evaden los obstáculos. A través de la fusión sensorial se logra mayor robustez que la alcanzada por los sensores individuales para detectar, identificar y evadir obstáculos. Así, obstáculos presentes en el ambiente de prueba como la escalera que no es detectada por el láser (0% de evasión) y el vidrio que la cámara no diferencia en la imagen (0% de evasión), el sistema los detecta, identifica y evade (100% de evasión).

El sistema de fusión desarrollado cuenta con módulos interconectados entre sí, que se encargan de recibir las medidas realizadas por el sensor láser y la cámara para extraer características que permiten identificar y etiquetar el obstáculo detectado. La etiqueta es utilizada para elegir la estrategia de evasión más adecuada según las características del objeto.

### Palabras clave

- fusión sensorial, cámara RGB, laser, detección y evasión de obstáculos.

## **ABSTRACT**

This document presents sensory fusion using two sensors: monocular camera that delivers images in RGB and a laser sensor that provides distance measurements in places. The result of the fusion is used for a mobile robot can perform navigation wandering in a structure environment test.

Navigation algorithms generally only detect and evade obstacles. Through the sensor fusion, it is possible give robustness to a system to detect obstacles that separate sensors can't. Also identify them and ensure their evasion. Thus obstacles in the test environment as the downstairs is not detected by the laser (0% evasion) and a camera no difference glass in the image (0% evasion), the system detects, identifies and evade (100% evasion).

The fusion system developed has modules interconnected with each other, which are responsible for receiving the measurements made by laser sensor and camera, later features are extracted allowing the system to identify by labeling the obstacle, and then use the label to choose the most appropriate navigation strategy to evade it.

## **Keywords**

- Sensor fusion, camera RGB, laser, detection and obstacle avoidance.

## TABLA DE CONTENIDO

1. INTRODUCCIÓN .....	1
2. MARCO TEÓRICO .....	3
2.1. INTRODUCCIÓN .....	3
2.2. ANTECEDENTES .....	4
2.2.1. Complementaria .....	5
2.2.2. Competitiva .....	5
2.2.3. Cooperativa .....	5
2.2.4. Híbrida .....	5
2.3. ARQUITECTURA DE BLOQUES DE FUSIÓN SENSORIAL.....	13
2.3.1. Celdas de fusión .....	15
2.3.2. Redes simples de fusión .....	16
2.3.2.1. Única celda de fusión .....	16
2.3.2.2. Redes paralelas .....	16
2.3.2.3. Redes seriales .....	17
2.3.2.4. Redes iterativas .....	18
2.3.3. Topología de la red .....	18
2.3.3.1. Descentralizada .....	18
2.3.3.2. Centralizada .....	19
2.3.4. Enfoque de la fusión sensorial .....	20
2.3.5. Reglas de fusión .....	20
2.3.6. Criterios para evaluar la fusión sensorial .....	21
2.3.6.1. No hay sensor infalible .....	22
2.3.6.2. El procesamiento capa superior - inferior, no evita los errores de la información que va de capa inferior - superior	22
2.3.6.3. La observación que genera el sistema de fusión puede ser mala respecto al mejor sensor del sistema .....	22
2.3.6.4. No hay algoritmos mágicos .....	22
2.3.6.5. Los datos de entrenamiento nunca serán suficientes ...	22
2.3.6.6. Dificultad para medir el valor de la fusión sensorial .....	23
2.3.6.7. La fusión sensorial no es un proceso estático .....	23
2.4. CONCLUSIONES .....	23
3. DESCRIPCIÓN DEL SISTEMA .....	24
3.1. INTRODUCCIÓN .....	24
3.2. CARACTERÍSTICAS DE LOS SENSORES .....	25
3.2.1. SENSOR LASER .....	25
3.2.2. CAMARA MONOCULAR USB .....	26
3.3. DESCRIPCION SOFTWARE .....	27
3.3.1. PROCESAMIENTO LÁSER .....	27
3.3.2. PRE-PROCESAMIENTO CÁMARA .....	30
3.3.3. PROCESAMIENTO CÁMARA .....	32
3.3.4. FUSIÓN SENSORIAL .....	34
3.3.5. DEFINICIÓN DE TRAYECTORIA .....	41

3.4.1. PLAYER .....	43
3.4.2. STAGE .....	44
3.4.3. OPENCV .....	45
3.5. COMUNICACIÓN PC - ROBOT .....	45
3.6. CONCLUSIONES .....	46
4. PRUEBAS Y RESULTADOS .....	48
4.1. INTRODUCCION .....	48
4.2. DESCRIPCION DE LOS ESCENARIOS DE PRUEBA .....	48
4.3. PRUEBAS DE NAVEGACION CON EL SENSOR LASER .....	49
4.3.1. Descripción .....	49
4.3.2. Laboratorio de Robótica móvil. Resultados y análisis ...	49
4.3.3. Segundo piso Escuela de Ingeniería Eléctrica y Electrónica Resultados y análisis .....	51
4.4. PRUEBAS DE NAVEGACION CON LA CAMARA .....	52
4.4.1. Descripción .....	52
4.4.2. Laboratorio de Robótica móvil. Resultados y análisis .....	53
4.4.3. Segundo piso Escuela de Ingeniería Eléctrica y Electrónica. Resultados y análisis.....	54
4.5. PRUEBA IDENTIFICACION USANDO FUSION SENSORIAL .....	57
4.5.1. Descripción .....	57
4.5.2. Resultados y análisis .....	57
4.6. PRUEBA NAVEGACION MEDIANTE FUSION SENSORIAL .....	59
4.6.1. Descripción .....	59
4.6.2. Resultados y análisis .....	60
4.7. ANÁLISIS GENERAL DEL SISTEMA.....	62
4.8. CONCLUSIONES .....	64
5. CONCLUSIONES .....	65
6. PERSPECTIVAS FUTURAS DE TRABAJO .....	66
7. REFERENCIAS BIBLIOGRÁFICAS .....	67
8. ANEXOS .....	71

## 1. INTRODUCCIÓN

Siempre que un robot interactúe con un ambiente, es necesario que cuente con algún nivel de percepción que facilitará su desempeño en diversas aplicaciones como: reemplazo del operador en tareas peligrosas, repetitivas o de precisión, asistencia a personas discapacitadas o en oficinas, tareas de búsqueda y rescate en desastres o emergencias (extendiendo los sentidos humanos), investigación en laboratorios o lugares agrestes, entre otras.

Los sensores que se conectan al robot están relacionados con la información del ambiente que necesita conocer el robot para desarrollar una tarea. Cada sensor tiene limitaciones que pueden superarse al combinar varios tipos de sensores e integrando la información que éstos proveen, lo que se conoce como fusión sensorial.

La fusión sensorial en robótica es amplia gracias a la diversidad de sensores que se pueden acoplar: de visión (cámara a bordo, vista de ave), de barrido (láser), sonares (eco, Doppler), de proximidad (inductivos, capacitivos), de contacto (bumper), entre otros.

En este trabajo de grado se explora la fusión sensorial entre un sensor láser 2D y una cámara monocular, de tal forma que se aproveche la precisión en las medidas de distancia del sensor láser y el amplio campo de visión de la cámara.

Aunque existen varias formas de realizar fusión sensorial, se eligió hacer uso del concepto de complementariedad para sacar provecho de las ventajas de un sensor cubriendo las limitaciones del otro y así obtener una información del ambiente que cada sensor por separado no podría obtener del sistema.

Este trabajo de grado fue guiado por el siguiente objetivo general:

Desarrollar un sistema de fusión de información entre un sensor láser y una cámara monocular para detección, clasificación y evasión de obstáculos durante la navegación de un robot móvil en un ambiente estructurado.

Para atender el objetivo general, se definieron cuatro objetivos específicos:

Obtener una revisión de los principales métodos de fusión sensorial entre sensores de diferentes tipos.

Definir el tipo de obstáculos a detectar por el sistema en un ambiente estructurado.

Implementar un algoritmo de fusión sensorial que integre la información por un sensor láser y una cámara monocular.

Definir y ejecutar un protocolo de pruebas para evaluar la potencialidad de la fusión sensorial desarrollada en la detección, clasificación y evasión de obstáculos durante la navegación móvil.

La organización del presente documento se hace en ocho capítulos, los cuales se mencionan a continuación.

El capítulo dos, marco teórico, se encuentran algunos de los antecedentes relacionados con la fusión sensorial, así como la teoría que sirvió de base para cubrir el objetivo general.

Luego en el capítulo tres, descripción del sistema, aparece el diagrama de bloques que describe la implementación del sistema de fusión sensorial y el software que se utilizó para realizarla.

Las pruebas y resultados se muestran en el capítulo cuatro, aquí se relacionan las pruebas realizadas para evaluar el desempeño de la fusión planteada y se presentan los resultados obtenidos, a partir de los cuales, se identifican los alcances y limitaciones del sistema desarrollado.

Por último, el capítulo cinco presenta las conclusiones generales, el capítulo seis las perspectivas de trabajo futuro y el siete las referencias bibliográficas utilizadas en el desarrollo del trabajo de grado.

En el anexo, se presentan algunos códigos de ejemplo que se utilizaron como base para la fusión sensorial, configuración del robot y conexión con los sensores.



## 2. MARCO TEÓRICO

### 2.1. INTRODUCCIÓN

Para que un robot móvil maniobre a través de un ambiente y realice alguna aplicación, debe cumplir con tres tareas clave: 1) detectar y esquivar los obstáculos en el ambiente, 2) medir la distancia entre los obstáculos, y 3) elegir una ruta segura [Zhan, Huang, Wu 08]. Las señales provenientes de los sensores en algunos casos se deben procesar, para extraer de ellas información relevante y suficientemente precisa para la realización de la aplicación que le ha sido asignada al robot.

Independiente de las técnicas o procedimientos que se aplique a las señales sensadas, el resultado en el desempeño del sistema debe mejorar en algunos de estos aspectos:

- **Representación:** Aumenta la información obtenida de las señales.
- **Certeza:** La incertidumbre en las señales del sensor disminuye.
- **Complejidad:** La información permite representar mejor el ambiente que rodea al robot.
- **Reducción de tiempo:** El tiempo necesario para procesar la información después del pre-procesamiento de las señales disminuye.
- **Robustez del sistema:** La presencia de disturbios no tiene efectos fuertes en el sistema de procesamiento de las señales.

Una de las formas para obtener esa mejoría es combinando la información proveniente de los sensores, ya sean de diferente o igual tipo (visión, distancia, contacto, etc.). Por lo tanto, cuando se combinan sensores aumenta la posibilidad de obtener información más completa y precisa, respecto a la que se obtiene con cada sensor por separado.

Sin embargo, surge la necesidad de definir la forma de integrar la información; en particular cuando la información proveniente de los sensores que difieren en dimensión, rango, alcance, precisión, entre otros aspectos relevantes. Es así como la fusión sensorial permite conciliar las diferencias señaladas anteriormente, además de la ambigüedad que se puedan presentar en los datos que integran la información de dos o más sensores.

La elección de los sensores a utilizar considera la aplicación que debe realizar el robot de tal forma que se facilite su interacción con el ambiente que lo rodea. En los robots móviles, está implícita la necesidad de navegación independientemente de la tarea que se espera ejecute; en este sentido, el presente trabajo usa un robot móvil que realiza navegación **wandering** en un espacio estructurado, haciendo uso de la fusión sensorial de dos de los sensores más usados en esa tarea: el sensor de barrido láser 2D y una cámara monocular. Se espera que mediante esta fusión el robot pueda incrementar la detección de obstáculos como vidrios, escaleras y otros objetos.

## 2.2. ANTECEDENTES

La tabla 2.1 resume las características más relevantes de algunos trabajos que han abordado la fusión sensorial frente al presente trabajo; posteriormente, se ampliará la información de los mismos.

Nombre del trabajo	Sensores usados	Aplicación de la fusión	Resultados obtenidos	Clasificación de la fusión [Mitchell 10]
<b>Fusion of laser and visual data for robot motion planning and collision avoidance. 2003</b>	2 Cámaras en configuración estereoscópica a 1 sensor láser.	Detección de obstáculos mediante distancia respecto al robot.	Detección y evasión de obstáculos aplicando la fusión al algoritmo <b>Vector Field Histogram</b> para navegación.	Cooperativa
<b>Panoramic 3D reconstruction by fusing color intensity and laser range data. 2006</b>	1 Cámara monocular. 1 sensor láser.	Generación de un mapa de profundidad a partir de las medidas de los sensores.	Reconstrucción de una parte de la escena usando vista panorámica.	Cooperativa.
<b>Face detection using Data and sensor fusion techniques. 2011</b>	Diferentes cámaras e imágenes de base de datos.	Detectar rostros en las imágenes para posterior reconocimiento en sistemas de seguridad.	Detección de rostros con 99% de acierto.	Complementaria
<b>Local Data fusion algorithm for fire detection through mobile robots.2013</b>	1 sensor de temperatura, 1 de intensidad lumínica, 1 cámara RGB.	Detectar y enviar alarma de conatos de incendio.	Reducción en la emisión de falsas alarmas y Confiabilidad en las alertas verdaderas.	Competitiva
<b>Identificación y evasión de obstáculos mediante fusión sensorial para robots móviles.2015</b>	1 Cámara monocular. 1 sensor láser.	Navegación <b>wandering</b> identificando obstáculos.	Buena identificación de obstáculos y evasión de los detectados.	Híbrida.

Tabla 2.1. Comparación entre trabajos. [Fuente: Propia]

La clasificación presentada en la tabla 2.1 está contenida en [Mitchell 10] y se hace de acuerdo a [Durrant – Whyte 88] que utiliza la configuración de los sensores usados en la fusión de datos, para catalogar el sistema.

Aunque existen otras formas de catalogar los sistemas de fusión, se elige esta puesto que es sencilla de aplicar conociendo la ubicación de los sensores y algunas (pueden ser pocas) características del sistema. Hay cuatro categorías las cuales se describen a continuación.

### 2.2.1. Complementaria

Se utiliza en entornos donde un solo sensor no puede obtener toda la información debido a su rango y alcance de operación. Así, los datos de los sensores (diferentes o iguales) se combinan para obtener una visión más completa del fenómeno que se desea observar o medir.

### 2.2.2. Competitiva

Esta organización se usa cuando el fenómeno a medir necesita de mayor calidad en su medida. Se caracteriza por la obtención redundante de una misma magnitud por varios sensores, los cuales compiten mostrando la medida con menos incertidumbre, error, o discrepancia, para ser elegida y pueda ser entregada al sistema de procesamiento. Al igual que en el caso anterior, no es necesario que sea el mismo tipo de sensor pero sí que se mida la misma magnitud usando el mismo o diferente método. Por ejemplo, cuando se calcula la posición angular mediante odometría y las mediciones de un giróscopo; se compara la odometría con el dato del otro sensor para verificar si el robot se movió X grados.

Su organización es compleja, debido a los posibles errores acumulativos o individuales de los sensores, y a que debe ser identificado aquel sensor que está fallando para que no afecte la competencia entre las medidas.

### 2.2.3. Cooperativa

Se caracteriza por combinar la magnitud de los sensores, para obtener una nueva magnitud que no sería posible obtener de cada sensor individual. Por ejemplo, la medida de profundidad a partir de dos cámaras monoculares organizadas en forma binocular. Aunque puede cambiar el tipo de magnitud obtenida, también acarrea incertidumbre y disminuye la precisión y seguridad de la nueva medida.

### 2.2.4. Híbrida

Es una organización flexible que permite utilizar las tres anteriores en distintas combinaciones según sea el uso final de la fusión:

- **Procesamiento sensorial de medidas:** Se refiere al proceso que se debe realizar en las medidas para garantizar mayor fiabilidad, precisión, mientras disminuye el error y la incertidumbre. (Se usa más con organización competitiva)
- **Procesamiento sensorial para el reconocimiento:** Se refiere al proceso que facilita al sistema la identificación de objetos en el rango de operación de los sensores, permitiendo asignarle etiqueta a los objetos que “ven” ellos. (Se usa mucho para la organización cooperativa)

Existen otras clasificaciones como la propuesta por [Moshe, Zhu, Kalata 97], que clasifica la fusión para robots móviles en **low level** y **high level**. **Low level fusion** se refiere a la integración directa de los datos con el objetivo de dirigir los movimientos del robot mediante comandos y/o señales de control a los motores. **High level fusion** se refiere a la integración indirecta de los datos mediante arquitecturas o módulos que permiten llegar desde el control de los motores hasta modelar el ambiente del robot. Así mismo, **low level fusion** tiene dos subdivisiones: las centralizadas y descentralizadas, de acuerdo con la forma en la se distribuye la realización de los subprocesos encaminados a la fusión.

A continuación, se hace una breve descripción de los antecedentes relacionados en la tabla 2.1:

**[Baltzakis, Argyros, Trahanias 03] Baltzakis, Haris. Argyros, Antonis. Trahanias, Panos. *Fusion of laser and visual data for robot motion planning and collision avoidance*.**

Utiliza un láser 2D con un rango de  $180^\circ$  para tomar medidas de distancia (una por grado), y dos cámaras en configuración estereoscópica. Se plantea la fusión de datos de distancia entre las medidas del láser y las de profundidad por estereovisión. Empieza con los datos obtenidos del ambiente por el sensor láser para generar un modelo en 3D, extendiendo en forma de línea los puntos de distancia en el eje Z (figura 2.1). Luego se toma una imagen I1 de una cámara C1 al mismo tiempo que se obtienen las lecturas del láser. Se halla a cada punto en la imagen su correspondiente en el modelo 3D para obtener una coordenada en (X, Y, Z). Después se toma otra imagen I2 por la otra cámara C2 buscando una equivalencia entre los puntos hallados en la I1 y los de I2 para validar el modelo 3D; si aparecen puntos que no tienen relación entre la extensión de líneas láser y los de las imágenes, se marca con una 'x' ese lugar (figura 2.2.B) siendo necesario aplicar geometría epipolar en las imágenes I1 e I2 para hallar esos valores de 'x' y ubicarlos en el modelo del ambiente fusionado (Figura 2.2.D).

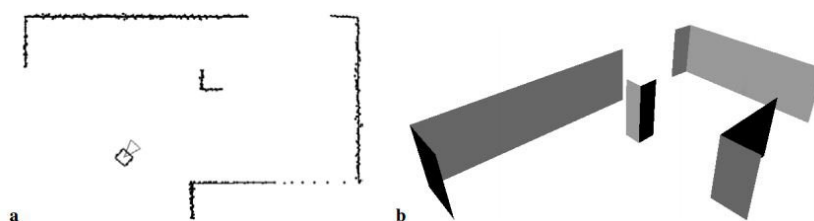
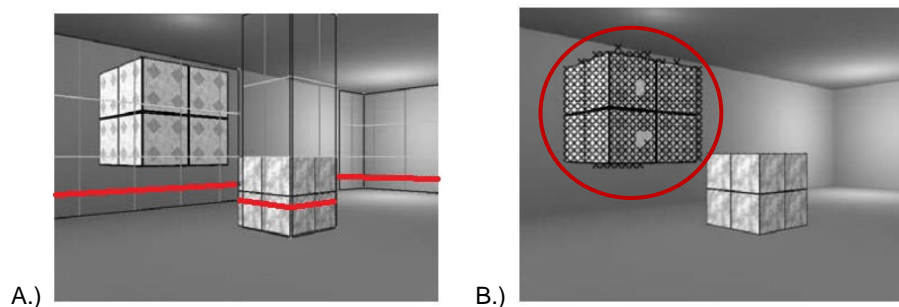


Figura 2.1. A) Imagen de la posición de robot y las lecturas (puntuales) del sensor láser. B) Extensión en líneas de las medidas del láser tomadas en a). [Baltzakis, Argyros, Trahanias 03]



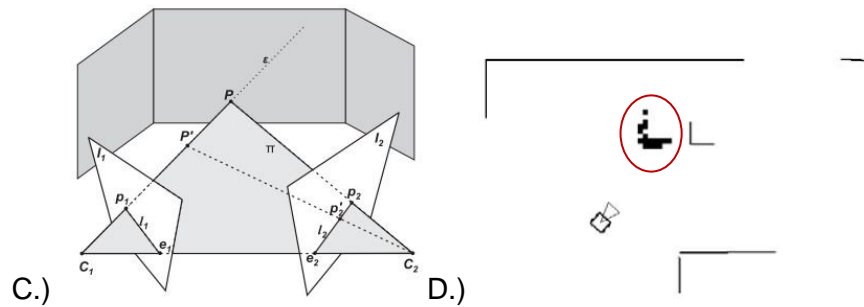


Figura 2.2.A.) Una escena de prueba, un cubo 'flota' 1m sobre el suelo y el otro está sobre el piso. La línea roja es el haz de sensor láser. B) Diferencias, encerradas en el círculo, y marcadas con 'x' entre el modelo 3D del láser y las imágenes de las cámaras en las zonas donde el haz láser no puede medir. C.) Posiciones de las cámaras para el cálculo de geometría epipolar. D.) Modelo fusionado del ambiente. El círculo indica el cubo que 'flota'. [Baltzakis, Argyros, Trahanias 03]

Una vez ubicados los obstáculos y las paredes del ambiente, se procede con la navegación usando el modelo del ambiente fusionado; esto se hace a través de una grilla de ocupación de obstáculos que utiliza el algoritmo VFH [Borenstein, Koren 91]. A partir de ésta, se determina una ruta libre y segura a través de la cual el robot puede transitar para llegar a un lugar de destino marcado en el mapa.

Los resultados obtenidos en las pruebas condujeron a las siguientes observaciones generales:

- Al usar **VFH** se necesita un mapa global, pero es suficiente con colocar sólo las paredes; no es necesario precisar la ubicación de los obstáculos.
- Las cámaras se ubican encima del sensor láser para cubrir una zona que el haz de este no detecta, para obtener una información que el láser por sí solo no puede entregar al sistema de fusión.
- Detecta obstáculos en el rango de fusión de los sensores; es decir, donde las cámaras contienen el haz del láser aunque son obstáculos estáticos.
- El eje Z del láser es diferente al eje Z de las cámaras. En el láser se refiere a la expansión en líneas de los puntos medidos, en la cámara a la distancia del objeto respecto a la misma.
- La configuración cooperativa de los sensores, permite al sistema de fusión detectar y evadir obstáculos tanto a 'nivel del suelo' como los que 'flotan' (figura 2.2) para eludir posibles colisiones con el robot.

[Wei, Jian 06] Wei Jiang, Lu Jian. *Panoramic 3D reconstruction by fusing color intensity and laser range data*. International Conference on Robotics and Biomimetics. 2006.

Este trabajo usa una cámara y un sensor láser para reconstruir una escena en vista panorámica, ubicando ambos sensores en una plataforma que gira dando vueltas completas a modo de escáner. Los sensores (cámara y láser) actúan de forma complementaria al ubicarse "de espaldas" uno al otro como se muestra en la Figura 2.3 (a 0° y 180° en la plataforma giratoria).

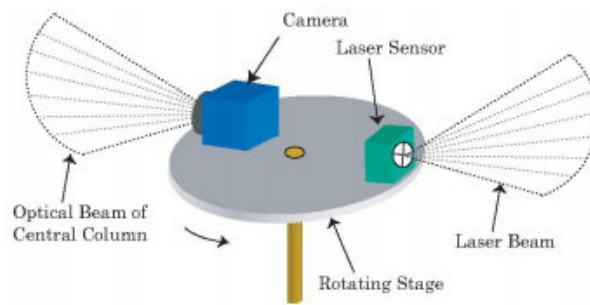


Figura 2.3. Plataforma que obtiene la vista panorámica de una escena observada. [Wei, Jian 06]

Para lograr una buena reconstrucción con el sistema de fusión es necesario: a) que no exista mutua occlusión para que la toma de medidas de un sensor no se vean bloqueadas por el otro, lo cual se logra con la plataforma giratoria, b) la información a fusionar tenga un mismo formato, y c) el cruce de los **FoV** de ambos sensores.

El sistema de fusión se divide en tres procesos: 1.) Medidas de profundidad del sensor láser 2.) Estimar profundidad usando técnicas de estereovisión. 3.) Fusión de la información.

El primer proceso se basa en el sistema de coordenadas y la relación geométrica de la figura 2.4. O denota el centro de la plataforma,  $r$  es la distancia entre el centro y el sensor láser,  $l$  se modela con la ecuación  $l = r + d \cos(\alpha)$ .  $\theta$  indica la posición del sensor mientras gira cuando toma la medida en el rango de  $[0 - 2\pi]$ . El ángulo  $\alpha$  se encuentra entre el rayo láser y el plano de la plataforma.

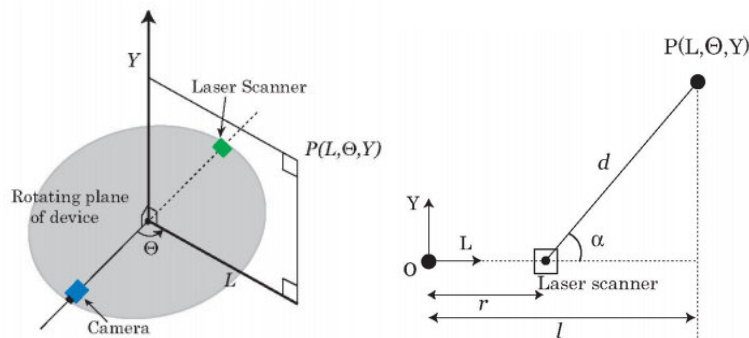


Figura 2.4. Sistema de coordenadas para el proceso 1. [Wei, Jian 06]

Este proceso entrega un vector de información  $P$  que contiene la distancia medida en esa línea  $L$ ,  $Y$ ,  $\theta$ .

El segundo proceso toma una imagen en cualquier ángulo de  $\theta$ , que sirve sólo como punto inicial para buscar las imágenes contenidas entre  $\theta$  y  $(\theta + - 90^\circ)$  las cuales se usan para construir un panorama en 3D (figura 2.5). El valor de distancia desde un 'pixel' en la 'imagen'  $(y, \theta)$  hacia el centro focal es  $z$ . Obtenido  $z$  a partir de estereovisión se aplican las ecuaciones 1 y 2.

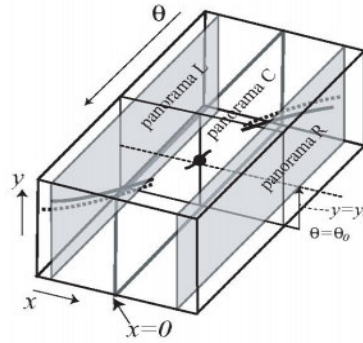


Figura 2.5 División de una imagen en panoramas. El panorama L es  $\theta - 90^\circ$  y R es  $\theta + 90^\circ$ . [Wei, Jian 06]

$$\text{Ecuación 1. } \theta = \theta_0 \pm \arctan \left( \frac{\zeta - \sqrt{\zeta^2 + \frac{(z+r)^2}{r^2 - 1}}}{1 - \zeta} \right) \quad \text{Ecuación 2. } \zeta = \frac{(z+r)^2 f}{r^2 x}$$

$$y = y_0 \times \frac{-xz}{f l \sin(\theta - \theta_0)}$$

Donde  $r$  es igual que en la figura 2.4 y  $f$  es la distancia focal. A cada 'pixel'  $(y, \theta)$  de la 'imagen' panorama L, R y de referencia en  $x = 0$ , se le halla su coordenada. Luego se compara entre el panorama de referencia - L, referencia - R, la intensidad de color de cada  $(y, \theta)$  y las diferencias entre cada par de panoramas se suma en **SSD (Sum of Squared Differences)**. Así se obtiene una disparidad para la L y otra para R.

El tercer proceso, la fusión, comienza con la información de los dos procesos anteriores y se aplica el enfoque de minimización que le asigna a un grupo Q de píxeles y P de medidas láser, el menor valor de etiqueta que corresponda a cada pixel de Q con una medida en P. Con esto, se conoce el costo de cada etiqueta y el cual se modela el costo como un **Markov Random Field (MRF)** el cual se optimiza usando **Bayesian Belief Propagation (BBP)** para obtener la fusión de las distancias.

La figura 2.6 presenta un ejemplo de reconstrucción en escala de grises de una escena real (figura 2.7) obtenida por la fusión, en donde se asignan los valores en nivel de gris de acuerdo con el costo para cada grupo Q y sus medidas P; entre más cerca del dispositivo, mayor nivel.



Figura 2.6. Observación final de la fusión. [Wei, Jian 06]

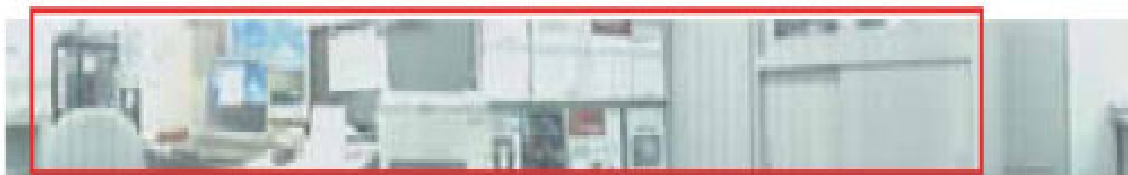


Figura 2.7. Escena real de la prueba realizada. [Wei, Jian 06]

Las pruebas realizadas permiten reportar en el artículo las siguientes observaciones:

- La oclusión entre los sensores se puede evitar mediante la construcción de una plataforma que los separa mientras permite tomar medidas.
- Se aprovechan las ventajas que tiene la cámara como: adquisición de la textura y amplio rango de visión (**FoV**); también las del láser que son: Precisión en las medidas de distancia y permite hallar las discontinuidades fácilmente.
- El sistema de fusión usa tres procesos para la reconstrucción, cuyos resultados se pueden usar por separado para comparar la fusión: sólo cámara con estereovisión, sólo láser con relaciones geométricas, fusión de datos con **MRF** y **BBP**.
- Los objetos deben estar estáticos, si se mueven pueden causar distorsiones a la reconstrucción final.

**[Jaisakthi, Chandrabose 11] Jaisakthi, S. M. Chandrabose Aravindan. *Face Detection using Data and Sensor fusion techniques*.**

El sistema desarrollado recibe una imagen; sea desde una cámara o procedente de una base de datos, con el objetivo de reconocer si contiene un rostro. Aplica cuatro métodos que extraen cuatro características diferentes las cuales se usan en dos técnicas de fusión distintas: fusión de datos e integración sensorial.

Los métodos usados son PCA (**Principal Component Analysis**), LM (**Legendre Moments**), ZM (**Zernike Moments**) y GFD (**Generick Fourier Descriptor**).

Para la técnica de fusión de datos, el resultado de cada método se concatena y se ingresa a modo de vector de características en una SVM (**Support Vector Machine**) que entrega como resultado una etiqueta que indica si la imagen contiene un rostro o no. La fusión sensorial (figura 2.8) se hace usando una **Machine Learning (ML)**.

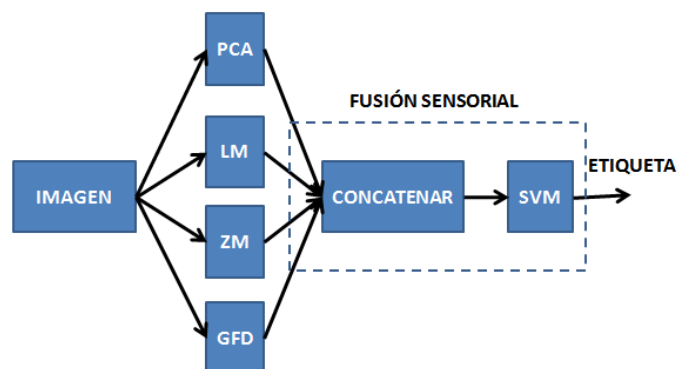


Figura 2.8. Diagrama de bloques de la fusión sensorial. [Jaisakthi, Chandrabose 11]

En cuanto a la integración sensorial, cada resultado de los métodos, ingresa a una SVM distinta la cual entrega una clasificación preliminar generando un vector con cuatro características que ingresa a una última SVM que emite una etiqueta si la imagen contiene un rostro o no. En este caso, la integración (figura 2.9) refina los resultados intermedios de manera separada para entregar una información más precisa.



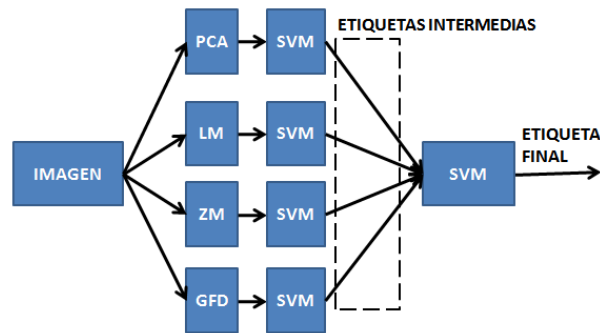


Figura 2.9. Diagrama de bloques de la integración sensorial. [Jaisakthi, Chandrabose 11]

Las pruebas se realizan con 500 imágenes (300 con rostros y 200 que no), utilizando el método de validación cruzada que consiste en dividir el total de datos en dos: un grupo de entrenamiento y otro de prueba mientras se cambia el contenido de imágenes en los grupos. Con esto, reportan en la tabla 2.2 los resultados de la validación.

Runs	PCA(%)		LM(%)		ZM(%)		GFD(%)	
	CV	Acc.	CV	Acc.	CV	Acc.	CV	Acc.
1	96.6	95.0	91	89	98	94.5	98.2	91.5
2	95.3	96.5	90	87	98	94.5	97.5	89.5
3	96.6	95	93	85	98	94.5	98	84.5
4	95.6	94	93	89	98.6	95.5	98.2	92.5
5	96.6	95	93	86	97	93	98	91.5
6	96.0	93.5	91	88	98	94.0	98.2	92.5
7	94.6	95.0	90	89	97.5	93.0	98.5	84.5
8	95.6	95.5	93	89	98.3	94.5	99	84.5
9	96.0	94.0	93	85	98.5	93.0	98.2	84.5
10	95.0	95	93	84	98	93.5	98	90.5
Avg	96.0	94.8	92	87.1	97.9	94.0	96.7	88.5

Tabla 2.2. Resultados obtenidos y comparados por cada método. [Jaisakthi, Chandrabose 11]

El grupo de entrenamiento es conformado por 180 de las 300 imágenes con rostros, y 120 de las 200 que no contienen, mientras que el de validación tiene el resto; 120 y 80. La columna Runs indica la iteración mientras se ejecuta la validación cruzada (**cross validation**). Cuando se empieza a integrar los resultados de dos, luego de tres y por último todos cuatro, se anota que el porcentaje de acierto sube: 97.8% (dos métodos usados), 99.2% (tres métodos) y 99.9% todos cuatro.

Las pruebas realizadas permiten reportar las siguientes observaciones:

- La combinación de distintos métodos de extracción de características aporta una precisión que puede ser refinada con la aplicación de procedimientos posteriores: sea fusión de datos o integración sensorial.
- Los procedimientos posteriores pueden disminuir la incertidumbre en los datos procesados.
- La fusión sensorial reduce la aparición de falsos positivos en los resultados y aumenta la cantidad de aciertos.

**[Roberto, Machado, Pinto, Branco 13] *Local data fusion algorithm for fire detection through mobile robot.***

La detección de conatos (inicio) de incendio es una señal importante en la prevención de daños estructurales en edificios, también en la preservación de bosques. En este artículo se propone una patrulla de robots con orugas para el movimiento todoterreno y que tienen un sistema de fusión usando datos procedentes de tres sensores diferentes: uno de temperatura (que mide en °C), otro de luminancia (mide en lumen) y una cámara (para el color de la llama).

Aunque existen otras técnicas como el índice Angstrom para detectar un conato de incendio, el artículo propone una fusión sensorial en forma competitiva para evitar las falsas alarmas; de modo que si un sensor envía la alarma, pero los otros dos no se activan, no se reconoce como incendio.

La técnica para hacer la fusión sensorial es la del promedio móvil. Se fija una ventana de tiempo de 10 segundos en la cual se toman y guardan las medidas de los tres sensores, luego se aplica el promedio y se envía como dato que se compara con un umbral para decidir si es o no una alarma.

Para fijar el umbral, la orientación del robot y conocer la distancia máxima/mínima para detectar un conato de incendio, se realizaron tres pruebas:

1.) Umbral de detección.

Se realizan medidas en los sensores cada 10 segundos y se observa el dato que sale como promedio. En el ambiente de prueba (interior de un edificio) el promedio fue: temperatura entre 20 °C – 35 °C, luminosidad entre 700 – 830 lumen. El umbral es 40°C para temperatura y 900 lumen para luminosidad.

2.) Distancia para detección.

Con los umbrales puestos, se coloca al robot enfrente de una llamarada y se detecta la llama por el color: amarillo (causada por papel, cartón, combustible) y azul (alcohol, químicos) a diferentes posiciones. Se determina que la máxima distancia es 1 m y la mínima es 10 cm para tener al robot seguro y sin peligro.

3.) Orientación del robot.

Se elige la distancia de 10 y 25 cm como distancia fija moviendo el robot en diferentes posiciones angulares para 'mirar' la flama.

Se determina que al frente de la llama se considera como 0°, a la izquierda se detectan hasta 40° y a la derecha hasta -40°.

Las pruebas realizadas reportan las siguientes observaciones:

- El sistema de fusión competitivo busca confirmar la alarma de uno con los otros sensores; así se evitan las falsas alarmas.
- La determinación de la distancia segura sirve para poder visualizar el conato de incendio. Muy lejos no detecta temperatura y luminancia, muy cerca se puede dañar.

- La técnica de promedio móvil de manera indirecta, reduce la incertidumbre en las medidas realizadas por los sensores.

El trabajo de grado titulado identificación y evasión de obstáculos mediante fusión sensorial para robots móviles utiliza la arquitectura de celdas de fusión descrita en [Mitchell 10].

Utiliza dos sensores: el sensor láser cuyas medidas de distancias sirven para detectar obstáculos además de la extracción de características en todo su rango. Y la cámara ubicada al frente del robot y encima del sensor láser, la cual tiene una inclinación hacia abajo de  $45^\circ$  cubre el espacio bajo el haz láser para detectar obstáculos como las escaleras hacia abajo.

El sistema de fusión hace navegación **wandering** así: busca seguir de frente mientras la ruta está libre de obstáculos, cuando detecta uno, lo identifica y evade.

Es de anotar, que la extracción de características busca rasgos de los obstáculos más comunes en el ambiente donde navega: pared, materas redondas, cuadradas, escalera.

### 2.3. ARQUITECTURA DE BLOQUES DE FUSIÓN SENSORIAL

En analogía con los robots, los seres vivos tienen sentidos que le permiten percibir el mundo que los rodea, con eso pueden tomar decisiones o alcanzar una meta. Sin embargo, ¿la información que se percibe es certera? Para ilustrar las ambigüedades que se pueden presentar, ¿la figura 2.10 corresponde a una copa o a dos personas?

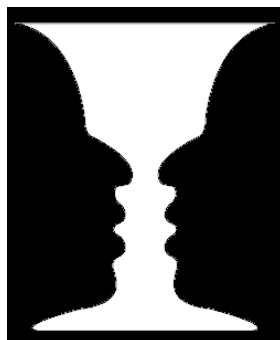


Figura 2.10. Información total, ¿dos personas de perfil o una copa?  
[\[http://keyseonel.wordpress.com/2010/09/20/percepcion-visual/\]](http://keyseonel.wordpress.com/2010/09/20/percepcion-visual/)

Por lo general, la elección de una opción para alcanzar un objetivo se hace de acuerdo con algún criterio.

En el caso de los seres vivos, la información provista por cada sentido se complementa con los otros para tomar decisiones. Cuando un robot necesita medir o “percibir” el ambiente que lo rodea, solicita datos a los sensores y ellos le informan

sobre: distancia, contacto, composición, forma, tamaño, todos datos por separado. Se busca a través de la fusión sensorial, entregar una información más completa.

La fusión sensorial es aplicada en varios campos donde se necesiten el uso de varios o diversos sensores, pero, independiente del campo de aplicación, la fusión sensorial se puede dividir en tres partes modulares para comprender el flujo de datos en el sistema (Figura 2.11).

Cada parte es un dominio que contiene los módulos más generales que intervienen en un sistema de fusión [Mitchell 10].

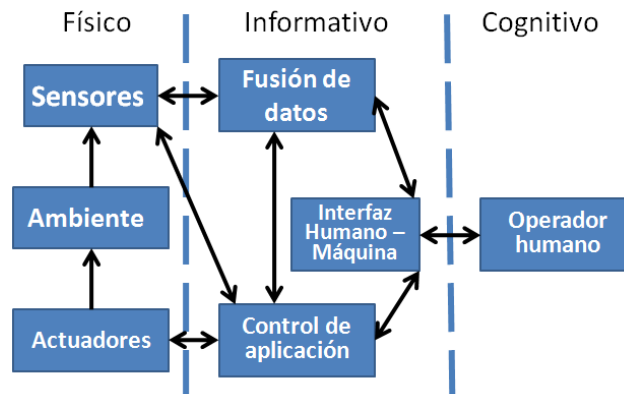


Figura 2.11. La división más general de los sistemas de fusión sensorial. [Mitchell 10]

**Dominio Físico:** Se refiere a la parte **hardware** del sistema de fusión. Reúne los dispositivos que hacen las medidas (sensores), los que alteran el contenido del ambiente o facilitan moverse en él (actuadores), y el ambiente mismo, ya que de este proceden las medidas del fenómeno que se quiere observar.

**Dominio Informativo:** Indica la parte **software** del sistema y es la parte más importante del mismo. El bloque de fusión de datos se construye a partir de módulos que ejecutan una función de fusión  $f$  que combina los datos procedentes de los sensores sea para obtener una medida más precisa y exenta de disturbio o para utilizarse en el control. El de control de aplicación también se encarga de administrar los recursos del sistema, además de reportar estados y/o recibir comandos del bloque IHM (Interfaz Humano Máquina), por otro lado, el control puede devolver los datos provenientes del bloque fusión si están incorrectos. El de IHM es la parte del sistema de fusión visible para el usuario, en algunos casos le permite enviar comandos al sistema, en otros solo puede leer los estados que le reportan.

**Dominio Cognitivo:** Es compuesta por el **usuario final**, que puede influenciar el sistema tomando decisiones y enviándolas como comandos en cada etapa, o bien como supervisor que corrige un reporte de estado incorrecto mientras el sistema se ejecuta de forma autónoma.

Debido a su estructura modular, el dominio cognitivo y el bloque IHM se pueden retirar sin impactar de forma desastrosa el sistema de fusión.

En cada dominio, la unidad más básica de procesamiento es la celda de fusión.

### 2.3.1. Celdas de fusión

También llamados **nodos**. Es la parte más sencilla en una red de fusión sensorial. Pueden encontrarse sea en la red o después de los sensores recibiendo todas las medidas que éstos realizan, para entregar una información que será usada en el resto del sistema.

Las entradas del nodo representan generalmente cualquier tipo de entrada y se agrupan en tres clases, las cuales son:

- **Información del sensor:** Se refiere a los datos que son enviados directamente del sensor o por otros nodos. Para la ecuación 3, es **Y**, **t**.
- **Información auxiliar:** Aquella que proviene de un procesamiento o se deriva de los datos del sensor. Para la ecuación 3, es **X**, **ΔY**.
- **Conocimiento exterior:** Son datos que se conocen *a priori*, provienen del exterior o que el usuario del sistema ingresa. En la ecuación 3, es **E**.

La nueva información se modela:

$$O_i = \langle E_i, x_i, t_i, y_i, \Delta y_i \rangle i \in \{1, 2, 3, \dots\} \quad \text{Ecuación 3.}$$

Donde:

**O** es la observación de salida del nodo.

**E** se refiere al nombre de la propiedad física que se está midiendo, y sus unidades.

**X** es la posición del sensor en el arreglo de sensores que envía la información.

La letra **t**, representa el momento de toma del dato.

**Y** es el dato de sensor **X**.

**ΔY** es la incertidumbre del dato **Y**.

El funcionamiento del nodo de fusión, se puede explicar mejor mediante sus propiedades:

1. **Estructura común:** Cada nodo tiene la misma estructura, lo que permite al diseñador colocarlos como módulos, o modificar su estructura.
2. **Integral:** El nodo tiene toda la información que necesita para trabajar en sus tareas.
3. **Pertinencia:** Cada nodo recibe la información que usa para generar una observación; nada más, nada menos.
4. **Incetidumbre:** El error en la observación y el ruido significan que la información en un nodo es incierta en mayor, o menor medida. Para asegurar

un grado razonable de seguridad de funcionamiento, hay que tener en cuenta esta incertidumbre en el tratamiento de la información de un nodo.

5. **Comunicación:** Los nodos sólo se pueden comunicar con aquellos que tengan conexiones directas.
6. **Auto-identificación:** La habilidad de cada nodo de identificarse a sí mismo en toda la red.
7. **Auto-inspección:** Proceso con el cual el nodo se asegura de su correcto funcionamiento y puede informar su estado a la red usando la variable E.

La figura 2.12 modela la interacción del nodo con los grupos de entradas y los resultados de salida llamados observaciones.

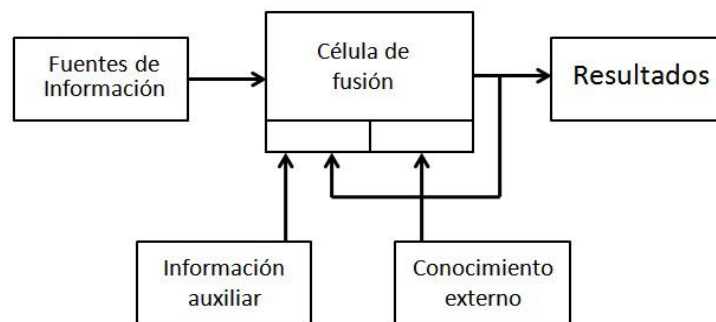


Figura 2.12. Gráfico de un nodo de fusión. [Mitchell 10]

### 2.3.2. Redes simples de fusión

Se agrupan las celdas (nodos) de fusión en redes multi-sensor, los cuales se pueden organizar en cuatro diferentes formas, sin necesidad de aumentar la complejidad de los nodos.

#### 2.3.2.1. Única celda de fusión

Es la más sencilla de todas las redes, se ubica después de los sensores para recibir la información sin procesar.

En esta red, la **variable E** hace funcionar la red como una supervisada. El modelo es igual a la figura 2.12.

#### 2.3.2.2. Redes paralelas

La siguiente en complejidad. Hace uso de la red anterior ubicando para cada sensor, una celda y su observación se convierte en entrada para una celda final que entrega una observación final, resultado de las anteriores fusiones.

Los resultados intermedios son los que se encargan de procesar los datos que reciben, la celda final se encarga de la precisión y confiabilidad para generar la observación final.

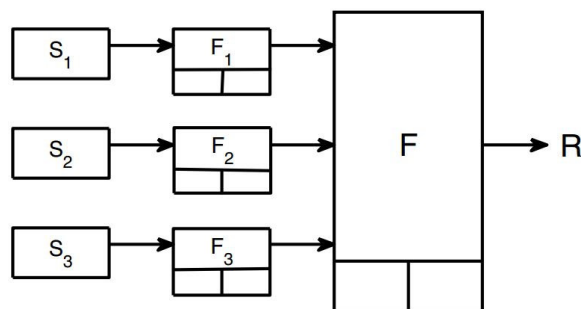


Figura 2.13. Gráfico de red paralela. [Mitchell 10]

La figura 2.13 muestra la organización de una red paralela, la letra S representa los sensores, F se refiere al nodo de fusión y R es la observación final.

### 2.3.2.3. Redes seriales

La tercera en complejidad. La organización se hace en cascada haciendo pares de sensores – celdas; la cantidad de entradas en la celda actual depende de la información intermedia que provenga de la celda anterior.

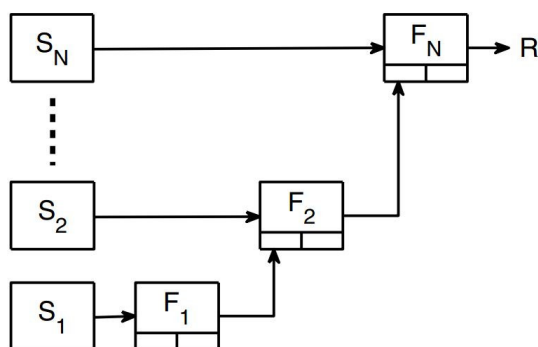


Figura 2.14. Gráfico de una red serial [Mitchell 10].

Debido a su organización, la fortaleza de esta red es la *complementariedad* de la *heterogeneidad*. Las entradas que se agrupan como información auxiliar, permite agregar más control o cambiar el procesamiento de los datos procedentes de los sensores y las observaciones de las celdas anteriores.

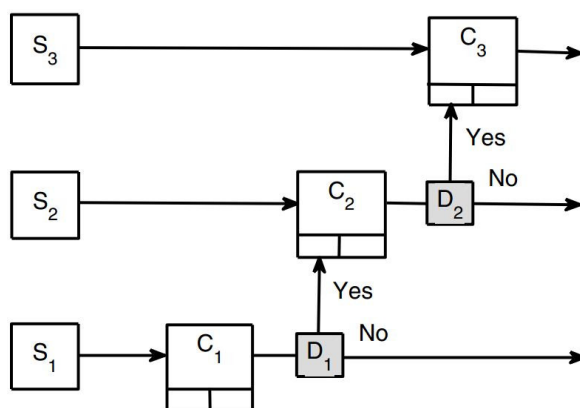


Figura 2.15. Gráfico de una red serial con clasificación de celdas [Mitchell 10].

Las figuras 2.14 y 2.15 muestran la organización de la red serial, se resalta que los resultados intermedios pueden entrar como información auxiliar (respecto a la ecuación 3) al siguiente nodo (figura 2.14) o pueden salir como observación hacia otro nodo o etapa de procesamiento del sistema (figura 2.15)

#### 2.3.2.4. Redes iterativas

La cuarta en complejidad. Hace una re-alimentación de la observación y entra como información auxiliar a la misma celda (Figura 2.16).

Esta organización se usa cuando el sensor o un procesamiento entrega datos dinámicos, donde la información del sensor  $Y$  es la que cambia en el tiempo  $t$ . Se usa en rastreo de objetos en imágenes, navegación en robótica móvil.

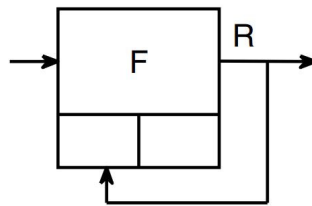


Figura 2.16. Gráfico de red iterativa [Mitchell 10].

La organización de los nodos de fusión no es excluyente; se puede utilizar diferentes redes según sea el caso. En este trabajo de grado se usa la red iterativa en el procesamiento de datos de la cámara, ya que en la búsqueda de continuidad en el piso se toma en cuenta el nivel de gris de la imagen anterior. Para el procesamiento láser se utiliza una red de única celda.

La información resultante del procesamiento del sensor láser y de la cámara deben llegar al nodo final: se elige la red paralela, porque la información se fusiona en el nodo final, no en uno intermedio o en el nodo que procesa a la cámara.

### 2.3.3. Topología de la red

Una vez tenemos la forma de organizar los nodos de fusión, se pasa a explicar las topologías que se indican cómo se comunican las redes de nodos. Además, se asignan las tareas y funciones a cada nodo para procesar la información que le llega. Se resalta que esta clasificación es respecto a las funciones.

#### 2.3.3.1. Descentralizada

Los datos de los sensores se procesan antes de llegar al nodo central por otros nodos auxiliares, esta es una característica que le da más fiabilidad en las observaciones que entrega el nodo final. Otras características son:

- **Comunicación:** Debido al procesamiento de los nodos auxiliares, solo se mueve la cantidad de información necesaria para que el nodo final funcione, evadiendo el cuello de botella.
- **Escalable:** Los nodos auxiliares pueden ser varios y desempeñar muchas funciones.



- **Robustez:** Tiene tolerancia a los fallos en las medidas o a las lecturas nulas.
- **Diseño modular:** Solo se debe programar qué hacer con los nuevos datos que entran, sin cambiar al resto.

Sin embargo, el uso de módulos no garantiza exclusividad en las lecturas de los datos que envían los sensores, haciendo que aparezcan problemas de redundancia y repetición de medidas. Para evitar esto, se debe programar los nodos auxiliares e indicar qué tipo de entrada serán los datos recibidos.

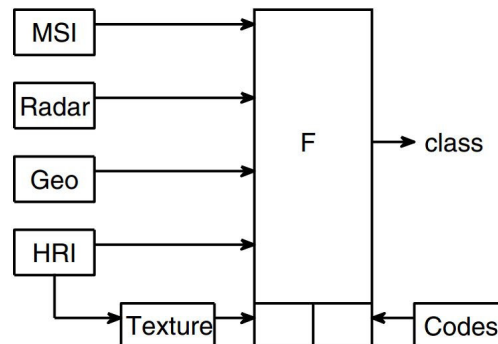


Figura 2.17. Ejemplo de arquitectura descentralizada [Mitchell 10].

La red paralela se usa mucho con esta topología, porque su organización facilita la separación de las funciones de los nodos. También se puede usar la de única celda asociándola con funciones de pre-procesamiento o especializadas.

### 2.3.3.2. Centralizada

A diferencia de la anterior, la fusión se realiza en un nodo central al cual llegan los datos procedentes de los sensores. Utiliza un único nodo que se encarga de preparar los datos para la fusión, realizar tareas de fusión y generar la observación final. Sin embargo, el uso de la menor cantidad de nodos no es garantía de un mejor desempeño.

Sus características son:

- **Comunicación:** Todos los sensores se comunican directamente al nodo central, lo que puede ocasionar cuellos de botella.
- **Rigidez:** Si la aplicación cambia la adaptación del nodo central no es rápida llegando incluso a la re-programación de este.
- **Vulnerabilidad:** Si el nodo central falla en cualquier parte, la observación final se afecta y ese efecto no se puede mitigar.
- **No modularidad:** Si la cantidad de sensores conectados cambia (agregar o quitar) se necesita re-programar el nodo central.

Otro tipo de topología es la jerárquica, la cual se puede consultar en [Mitchell 10].

En vista de las características, se elige la topología descentralizada porque la comunicación entre los nodos sólo mueve la información que necesita, usa un diseño modular de funciones permitiendo asignar funciones extras a nodos diferentes.

La construcción de funciones se hace mediante la heurística en la etapa de toma de datos de este trabajo de grado, luego se evalúa su desempeño con otras sesiones de toma de datos observando sus resultados para modificar o cambiar la función si no es satisfactoria o asignarla al nodo de fusión en caso afirmativo.

#### 2.3.4. Enfoque de la fusión sensorial

Una vez se tiene la forma de organizar los nodos, la asignación de las funciones, se presenta la finalidad del sistema de fusión.

El enfoque da cuenta sobre el objetivo que se persigue:

- **Enfoque hacia los datos:** Datos más precisos, libres de incertidumbre y/o que se puedan utilizar más fácilmente sin aumentar su procesamiento.
- **Enfoque hacia el reconocimiento:** Generar una observación a partir de los datos recibidos que busque una interpretación que reconozca lo mejor posible el objeto que “ve”.

El enfoque en este trabajo de grado, es hacia el reconocimiento porque no se va a controlar directamente los motores del robot y se quiere llegar a reconocer algunos de los obstáculos más comunes en el ambiente de prueba mediante la geometría que posean.

#### 2.3.5. Reglas de fusión

Es una de las múltiples formas de asignar una función de fusión  $f$  a un nodo como se puede leer en [Moshe, Zhu, Kalata 97] y [Luo, Kay 89]. La función de fusión es el núcleo de todo sistema de fusión sensorial y algunas son excluyentes (única función en todos los bloques de fusión) mientras que otras no (diferentes funciones en cada bloque).

Una de las funciones no excluyentes y flexibles, es la descrita en [Flynn 88] la cual expone un ejemplo de la técnica de reglas de fusión, cuya metodología consiste:

- En pro de determinar la mejor forma de fusionar la información entregada por múltiples sensores, éstos se deben caracterizar para observar sus fortalezas y debilidades mediante procesos de toma de datos.
- Definir un formato para la entrega de datos de los sensores que permita ver las fortalezas de ellos.
- Comparar los datos de ambos sensores y examinar las zonas donde se cruzan y las que no para observar información relevante. Extraer las reglas de fusión.
- Realizar pruebas con las reglas de fusión planteadas y verificar los resultados. De ser necesario se ajustan, crean o modifican las reglas dependiendo de la

aplicación en donde se usará la fusión sensorial.

- Se compara la fusión sensorial con el uso de un solo tipo de sensor mediante pruebas. Si la fusión es satisfactoria se entrega la información para que el robot cumpla con la aplicación asignada.

### 2.3.6. Criterios para evaluar la fusión sensorial

La fusión sensorial busca mejorar la calidad de los datos, extraer más información, permitir mejorar la capacidad de respuesta de un sistema, llegar hasta el reconocimiento de objetos, entre otros objetivos.

Aunque tiene todas esas ventajas, hay algo que no se debe olvidar y se busca evadir siempre: **la fusión negativa [Liggins, Hall, Llinas 09]**, ya que aparece cuando la observación final del sistema es errónea.

Para detectarla, se evalúa el desempeño del sistema con la fusión sensorial (sea de datos o de reconocimiento) y se compara con el desempeño del sistema sin la fusión usando directamente los sensores aplicando la ecuación 4.

$$p_F \ll \min_m(p_m) \quad \text{Ecuación 4.}$$

Donde  $p_F$  es el desempeño del sistema, y  $p_m$  es el desempeño del sensor  $m$  del sistema. Al cumplirse lo que indica, hay una evidencia de que la fusión aplicada no es correcta ni aporta mejoría alguna.

También se puede evaluar la fusión sensorial con el ambiente usando la ecuación 5.

$$E_F = E_1 \wedge E_2 \wedge E_m \quad \text{Ecuación 5.}$$

En la ecuación 5, el ambiente  $E_m$  (**Enviroment**) describe el lugar donde cada  $m$  sensor trabaja. Para saber el funcionamiento del sistema  $E_F$  se hace interceptando cada ambiente  $E$  de los  $m$  sensores.

La información que se cruza son los datos que el sensor entrega o las características de este: medir distancia, resolución, otras.

Para detectar la fusión negativa con la ecuación 5, hay una condición:

- Si el ambiente  $E_F$  es inconsistente con alguno de los  $E_m$  (intercepción vacía), ese sensor  $m$  es dominante y puede inclinar las medidas a su favor como si fuera el único que entrega datos.

El efecto de la fusión negativa en cada topología de red es:

- En una arquitectura centralizada no se puede mitigar ese efecto fuera del nodo central, aumentando las funciones que ese nodo debe asumir.
- La arquitectura descentralizada puede delegar al nodo auxiliar encargado de ese sensor dominante, la acción de mitigación en la influencia de los resultados intermedios que este entrega al nodo de procesamiento.

- La arquitectura jerárquica permite usar toda la información de ese sensor evitando la dominación sin necesidad de reducirla debido a la jerarquía que el diseñador le asigne.

Comparando los resultados de la fusión con la realidad, dará cuenta de la efectividad de la arquitectura de fusión sensorial. Con las ecuaciones 4 y 5, se tiene la certeza de evadir la fusión negativa, aunque existen otras características que no se pueden medir como las que se describen a continuación.

#### **2.3.6.1. No hay sensor infalible**

Esto significa que si hay algo que no puede ser medido, observado o inferido de los datos recolectados desde el sensor, no hay fusión que pueda generar esa observación.

#### **2.3.6.2. El procesamiento capa superior – inferior, no evita los errores de la información que va de capa inferior – superior**

Se debe buscar hacer el mejor tratamiento posible en cada paso del proceso de fusión. Si no se realiza el procesamiento de datos o falla la selección y refinamiento de las características apropiadas y eficaces, el error no puede ser cubierto por la elección de las técnicas complejas de reconocimiento de patrones. Simplemente debemos prestar atención en cada etapa de la cadena de información, desde la detección de la información hasta la creación de la observación.

#### **2.3.6.3. La observación que genera el sistema de fusión puede ser mala respecto al mejor sensor del sistema**

La falta de evaluar con precisión la exactitud de los datos del sensor / entrada llevaría a sesgos y errores en los resultados fusionados o la información que se usa para generar la observación se bloquea entre sí (oclusión, mutua exclusión).

#### **2.3.6.4. No hay algoritmos mágicos**

Existen varios algoritmos y técnicas para la fusión sensorial. Aunque hay algoritmos cada vez más sofisticados, es siempre un desafío que coincida con el objetivo del diseñador para poder hacer lectura de los datos, desarrollar un sistema, y hacer inferencias.

#### **2.3.6.5. Los datos de entrenamiento nunca serán suficientes**

Es bien conocido que los métodos de reconocimiento de patrones, tales como las redes neurales, requieren datos de entrenamiento para establecer los pesos sinápticos, clave en el funcionamiento de la red neuronal. Los métodos híbridos que combinan la información implícita y explícita pueden ayudar, y los nodos tienen la capacidad de recibir ambas informaciones de acuerdo a la ecuación 3.

Por ejemplo, cuando se busca asignar un vector de características  $n$  dimensional a uno de  $m$  clases o categorías, necesitamos en general  $n \times m$  ejemplos de entrenamiento bajo una variedad de condiciones de observación. Esto puede ser muy difícil de obtener, debido a la forma dinámica de los datos.

#### **2.3.6.6. Dificultad para medir el valor de la fusión sensorial**

Debido a que se empieza a valorar la fusión sensorial desde la efectividad de los algoritmos (lado del sensor), y después se toma en cuenta la interpretación de esas observaciones, el operador no interviene en la construcción del sistema. Es el diseñador que debe convertir los requisitos del operador en objetivos del sistema, haciendo que la fusión se evalúe desde el resultado: sirve o no sirve.

#### **2.3.6.7. La fusión sensorial no es un proceso estático**

Esto se debe a la cantidad de sensores que hay actualmente, las formas de medir las magnitudes físicas, las distintas formas de comunicarse, la abundancia de dispositivos, el alcance de internet en la vida de las personas, el procesamiento y almacenamiento de la información, hace que los métodos de fusión siempre estén en constante cambio.

### **2.4. CONCLUSIONES**

La revisión bibliográfica en temas de fusión sensorial que involucran cámaras de visión y laser evidenció una amplia gama de aplicaciones entre las cuales se destacan la reconstrucción de escenas observadas, la navegación en robots móviles, la búsqueda y reconocimiento de objetos entre otros. Las áreas donde se involucra la fusión sensorial igualmente son múltiples y variadas, donde sobresalen por su nivel de desarrollo e impacto en la investigación la militar, la médica y la robótica.

- La aplicación de cualquier técnica de fusión tiene por lo general cinco etapas: toma de datos, formulación de la fusión, ejecución de la técnica formulada, evaluación de los resultados, decisión: si es satisfactoria se entrega la observación hacia la aplicación de lo contrario se devuelve a la formulación de la fusión.
- Para la fusión de datos es necesario buscar alguna similitud que permita la comparación y el contraste entre datos, y así llegar a tener más certeza en las medidas o generar otra (enfoque de datos).
- El concepto de celda de fusión posee flexibilidad y se puede adaptar a cualquier tipo de aplicación porque se pueden utilizar distintas funciones de fusión sean o no excluyentes.
- Se debe resaltar que los criterios para evaluar la fusión sensorial son características de la fusión catastrófica y de otras que no se pueden medir, porque afectan al sistema de fusión en general ya que pueden influir en cada celda de fusión del sistema; también los errores se pueden volver acumulativos afectando desde la etapa donde no se corrigen hacia aquellas que tenga conectadas.

### 3. DESCRIPCIÓN DEL SISTEMA

#### 3.1. INTRODUCCIÓN

La figura 3.1 presenta el diagrama de bloques del sistema propuesto para lograr la fusión sensorial, y a partir de ella, identificar obstáculos que modifican, para que el robot móvil navegue de manera segura.

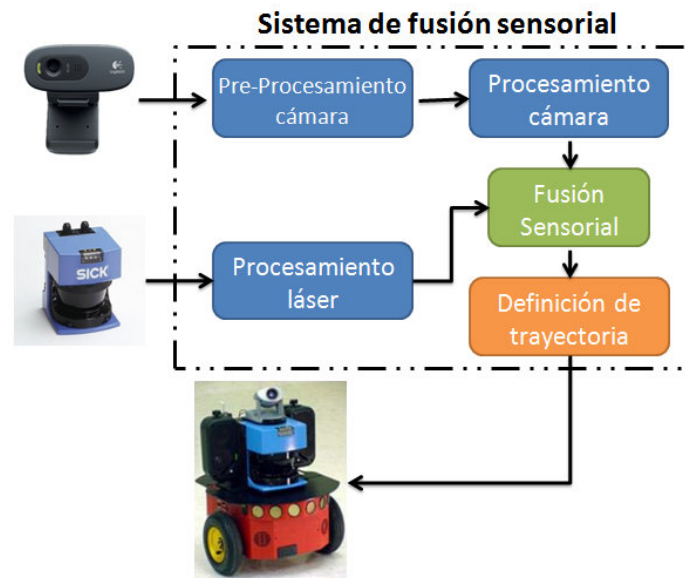
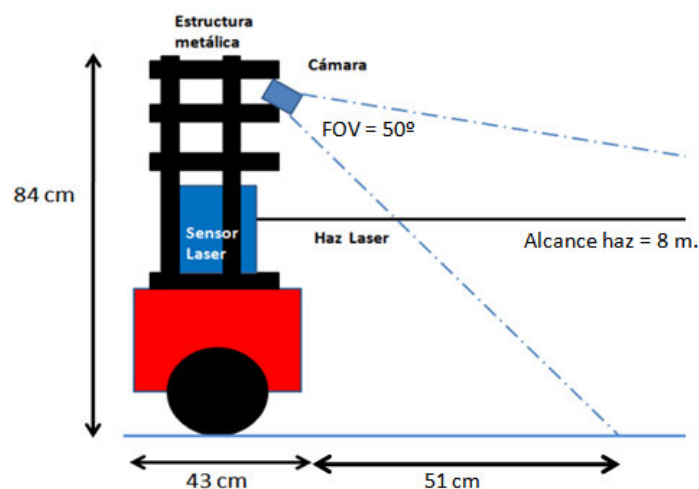


Figura 3.1. Diagrama de bloques del sistema desarrollado. [Fuente propia]

El sistema tiene dos entradas: una cámara monocular a bordo y un sensor láser SICKLMS 200, que entrega 361 medidas de distancia realizadas durante su barrido entre  $0^\circ$  y  $180^\circ$  (una medida por cada  $0.5^\circ$ ). La figura 3.2 muestra las dimensiones del robot, también la posición del láser y la cámara (con una inclinación hacia abajo aproximadamente  $45^\circ$ ) en el robot.



3.2. Zonas de funcionamiento de los sensores en el robot. [Fuente propia]

El bloque de procesamiento láser realiza tres tareas: 1) resaltar las diferencias entre las medidas, 2.) obtenerlas características geométricas en los obstáculos detectados y 3) identificar la probabilidad de ocupación en un conjunto de 36 regiones alrededor del robot.

Por otro lado, la cámara entrega una imagen RGB de 640x480x3 que ingresa al bloque de pre-procesamiento cámara en donde: 1.) se reduce el tamaño de imagen a la mitad sin pérdida relevante de información, 2.) se transforma de RGB a YIQ, y 3.) se aplica filtros pasa-bajo al plano Y para disminuir cambios fuertes en el nivel de gris de los píxeles favoreciendo la detección del piso.

Llegada la imagen del bloque anterior, el bloque procesamiento cámara busca la continuidad del piso utilizando la técnica del histograma, en pro de facilitar la extracción de características de un obstáculo en particular debido al alto riesgo para el robot cuando lo encuentra: las escaleras hacia abajo.

La integración de las características obtenidas usando el sensor láser con las conseguidas empleando la cámara se realiza en el bloque fusión sensorial para identificar el obstáculo en el rango donde se cruzan los FoV de ambos sensores, y determinar los estados (libre/ocupado) de posibles trayectorias para el robot.

Por último, el bloque definición de trayectoria elige la ruta más segura en base a: 1) criterios de riesgo (cercanía al robot), y 2) el tipo de obstáculo. Una vez definida la ruta se envían los comandos de movimiento para realizar navegación.

La salida es un comando que le indica al robot si el movimiento es de traslación (valores entre 0.0 y 1.0) o rotación ( $0^\circ$  -  $180^\circ$ ) además de su sentido positivo (marcha adelante/giro a la izquierda) o negativo (reversa/giro a la derecha). El robot utilizado en este trabajo de grado es la plataforma Pioneer 3DX [MobileRobots 06].

### 3.2. CARACTERÍSTICAS DE LOS SENSORES:

#### 3.2.1. SENSOR LÁSER [SICK AG Division].

El sensor láser utilizado es el SICKLMS 200 cuyo principio para medir distancia es el **tiempo de vuelo** (figura 3.3) del pulso láser: un pulso sale desde el dispositivo y cuando toma contacto con una superficie, retorna y el tiempo que se demora en ir y volver lleva consigo la información de distancia desde el dispositivo hacia la superficie.



Figura 3.3. Forma en la que el sensor láser toma medida de distancia por el método de tiempo de vuelo del haz láser [SICK AG Division].

Cuando el haz no rebota en ninguna superficie, se asigna la medida máxima configurada. La configuración utilizada mediante el software Player, en este trabajo de grado fue:

- Rango del haz láser: entre  $0^{\circ}$  –  $180^{\circ}$ , aunque se puede configurar (**FoV**) en otros valores, se elige usar el máximo para una mayor cobertura.
- Cantidad de muestras: 181 ó 361 en el **FoV** configurado, se utiliza 361 para tener más cantidad de datos.
- Resolución angular: valores entre  $0.25^{\circ}$ ,  $0.5^{\circ}$ ,  $1^{\circ}$ . Se usa  $0.5^{\circ}$  para corresponder los ángulos con las 361 muestras.
- Alcance: mínimo de 0.05m hasta un máximo de 80 m. El que se usa en el trabajo de grado es un máximo 8 metros ya que el ambiente es **indoor**.

La siguiente tabla muestra otras características del láser.

Característica	Valor
<b>Alcance</b>	Máximo 80 metros.
<b>Resolución angular</b>	$0.25^{\circ}$ , $0.5^{\circ}$ , $1^{\circ}$ para elegir.
<b>Tiempo de respuesta</b>	53 ms, 26 ms, 13 ms
<b>Resolución de la medida</b>	10mm
<b>Error en la medida (buena visibilidad, T = <math>23^{\circ}\text{C}</math>)</b>	Típico de + - 15 mm (medida en mm, alcance de 8 m) o de + - 4 cm (medida en cm)
<b>Error estadístico</b>	Típico de +- 5 mm (medida en mm, alcance de 8m)
<b>Interface de datos</b>	RS 232/ RS 422
<b>Tasa de transferencia</b>	9.6/ 19.2/ 38.4/ 500 kBaud

Tabla 3.1. Características del láser. [SICK AG Division]



Figura 3.4. Foto del sensor láser. [SICK AG Division]

### 3.2.2. CÁMARA MONOCULAR USB

Se utiliza una cámara Logitech HD C270 que cuenta con el driver UVC, un campo de visión (**FoV**) de  $60^{\circ}$  y conexión USB. Las características de la cámara son:

Característica	Valor
<b>Resolución de video</b>	HD 720p
<b>Enfoque</b>	Automático
<b>Corrección de luz</b>	Automática usando tecnología Fluid Crystal
<b>Tamaño imagen</b>	1280 x 720 pixeles, máximo.
<b>Conexión</b>	Puerto USB 2.0 y driver UVC.

Tabla 3.2. Características de la cámara.





Figura 3.5. Cámara utilizada.

### 3.3. DESCRIPCIÓN SOFTWARE DEL SISTEMA

Todos los bloques de la figura 3.1 corresponden a desarrollos en software. Estos fueron implementados en el lenguaje de programación C++ debido a la compatibilidad con el programa Player, que controla al robot Pioneer 3DX, y con las librerías de OpenCV, que se encargan de procesar las imágenes.

Para describir el flujo de datos e información entre los sensores y los bloques se usa la ecuación 3, respecto al flujo entre los bloques y por último el robot, se utiliza la nomenclatura de la figura 2.12. Tanto la ecuación como la figura son descritas en el capítulo 2.

#### 3.3.1. PROCESAMIENTO LÁSER

Un bloque de pre-procesamiento no se hace necesario debido a las características del sensor láser (Tabla 3.1), que tiene un error estadístico de 5mm para un alcance de 8 m en las medidas de distancia que toma. Este error se considera tolerable ya que los obstáculos tienen medidas en cm y los espacios libres se asocian con las distancias más grandes, en m.

Inicialmente, este bloque recibe el conjunto de 361 lecturas (una cada  $0.5^\circ$  en el rango de  $0^\circ - 180^\circ$ ) obtenidas durante el barrido del sensor que se realiza en un tiempo de 100ms.

Utilizando la ecuación 3,  $X = 1$  indica que el láser es el primer sensor que envía información al sistema de fusión sensorial,  $E$  = distancia en metros, en el programa se almacenan en un vector tipo **double**,  $Y$  = medida de distancia que se recibe como lectura (recopiladas en el vector **Y**), la incertidumbre del dato ( $\Delta Y$ ) no se toma puesto que se considera tolerable y el momento de captura de la medida ( $t$ ) no se usa porque sólo se recibe el último barrido completo que hace el sensor al momento de solicitar lecturas mediante Player, ya que se considera el más actualizado.

Las medidas que hace el sensor son en metros, y los datos contienen siete cifras significativas. En pro de facilitar la primera tarea (Resaltar las diferencias entre las

medidas) se usan tres cifras para que la unidad sea milímetros multiplicando cada  $Y$  x 1000.

La figura 3.6 muestra el rango y la forma del haz láser, además de un ejemplo de medición en un ambiente simulado en Stage.

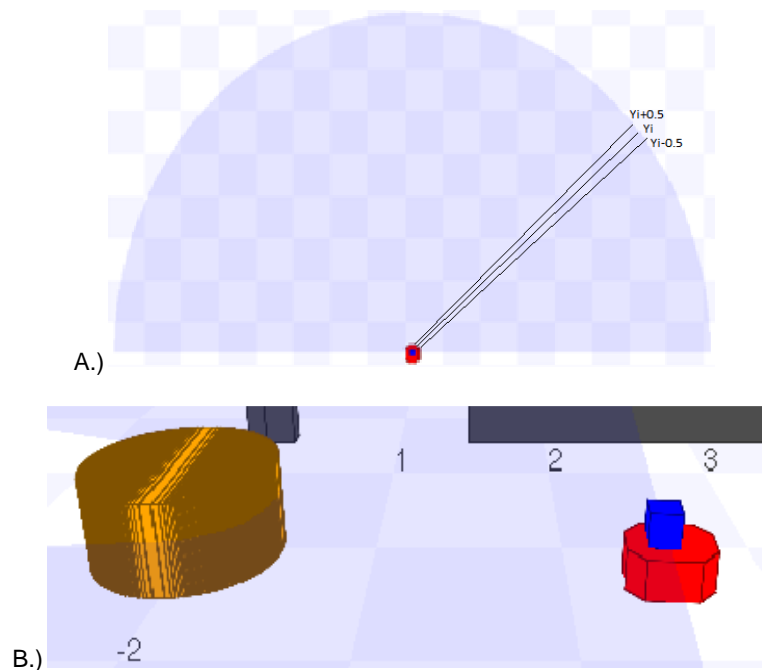


Figura 3.6. A.) Simulación en Stage. Arriba se muestra el haz del láser y su disposición en forma polar. B.) se muestra el haz láser realizando una medida de distancia de un objeto. [Fuente propia]

Este bloque efectúa tres tareas: La primera se realiza tomando cada medida  $Y_i$  del sensor y restando ese valor con la siguiente  $Y_{i+0.5}$  (Figura 3.6.A) obteniendo tres posibilidades según la siguiente ecuación 6:

$$Y_{i+0.5} - Y_i = L_j \quad \begin{matrix} i \in [0^\circ, 0.5^\circ, 1^\circ, \dots, 179.5^\circ, 180^\circ] \\ j \in [0, 1, 2, 3, \dots, 359, 360] \end{matrix} \quad \text{Ecuación 6.}$$

$$\begin{aligned} M_j &= 4, & \text{si } L_j < 0 \\ M_j &= 2, & \text{si } L_j > 0 \\ M_j &= 8, & \text{si } L_j = 0 \end{aligned}$$

Ya que esta resta corresponde a una forma de calcular la derivada de las medidas de distancia, el resultado es un vector  $M$  que corresponde a la pendiente a través de la cual se puede empezar a distinguir entre un obstáculo y otro, buscando cuándo ocurre un cambio de pendiente, utilizando la ecuación 7.

Sin embargo, es necesario agregar otra característica para facilitar y delimitar mejor el contraste entre obstáculos, puesto que puede haber algunos adyacentes que tienen igual pendiente, pero están separados por una distancia.

Para detectar esa distancia, se aprovecha la sustracción entre las medidas  $Y$  del láser buscando algún cambio igual o superior a 20 cm. Con estas dos características, la distinción entre conjuntos de medidas que permiten obtener una agrupación la cual facilita la clasificación de los obstáculos es contenida en un vector  $V$ , usando la ecuación 7.

$$\begin{aligned} V_k &= M_j, \text{ si } M_j \neq M_{j+1} \\ V_k &= L_j, \text{ si } |L_j| \geq 200 [mm] \\ j &\in [0, 1, 2, 3, \dots, 360] \quad k = k + 1 \end{aligned} \quad \text{Ecuación 7.}$$

El vector  $V$  contiene las referencias que indican el inicio y final de cada grupo de medidas  $Y$  las cuales se asocian a un obstáculo. Este vector es de tamaño variable, y aunque el ambiente es estructurado, se desconoce la cantidad de obstáculos que contiene.

Con lo anterior, se tienen listos los datos para realizar la siguiente tarea.

La segunda tarea se encarga de preparar los datos para extraerles tres características: el ángulo que describe la forma del obstáculo, la pendiente que tienen los puntos sucesivos dentro del grupo y el porcentaje de puntos bajo los extremos. La extracción se describe en el bloque fusión sensorial.

Aunque el sensor posee un rango entre  $0^\circ - 180^\circ$ , los obstáculos que pueden detener al robot son los que se ubican al frente de este, por tal razón se restringe la búsqueda de características al rango  $60^\circ - 120^\circ$ , y aquellos detectados en ese rango son contenidos en un vector  $V1$ .

Las características se pueden extraer aplicando la recomendación descrita en [Xavier, Pacheco, Castro, Ruano, Nunes 04], la cual explica que la detección de objetos se puede facilitar mediante la comparación de características. Por ejemplo, en el ambiente donde el robot hace su navegación existen obstáculos como: materas, mesas, paredes, personas, vidrios, puertas, botes y canecas para residuos. Estos obstáculos se pueden separar de acuerdo a su geometría en dos grupos: redondos (materas, piernas de personas, botes y canecas para residuos) y planos (paredes, puertas, vidrios, patas de mesa). Todos a la altura del haz láser.

La primera característica se extrae aplicando el método **Inscribed Angle Variance (IAV)** descrito en [Xavier, Pacheco, Castro, Ruano, Nunes 04] el cual entrega un ángulo entre  $90^\circ$  y  $180^\circ$  como resultado final. También se añade la varianza para filtrar posibles errores que conduzcan a clasificaciones equivocadas.

La segunda característica se extrae midiendo la pendiente entre las medidas adyacentes  $Y_i - Y_{i+0.5}$  después de hacer una descomposición vectorial de la medida entregada por el sensor láser y aplicando la ecuación geométrica de la pendiente. Además, se usa la varianza para refinar el resultado y disminuir la incertidumbre en esa cifra.

La tercera característica se calcula sacando una referencia de los puntos extremos de cada grupo de medidas, es decir el  $Y_{inicial}$  y el  $Y_{final}$ ; la referencia es el menor valor entre ellos. Luego se cuentan cuántos puntos  $Y$  son menores que la referencia generando un porcentaje el cual muestra una proporción que se puede asociar con la curvatura del obstáculo.

Con estas tres características se pueden distinguir los obstáculos redondos de los planos. Sin embargo, con estas dos categorías, no es suficiente la identificación de todos los obstáculos presentes en el ambiente donde el robot se mueve. No obstante, en el bloque fusión sensorial se añaden otras para generalizar todos los obstáculos que pueden aparecer dentro del ambiente estructurado.

La última tarea se hace en base al algoritmo **VFH [Borenstein, Koren 91]**, agrupando las medidas del sensor en conjuntos de diez, de ahí que se formen las 36 regiones.

La tarea empieza a buscando la distancia crítica en cada grupo.

La distancia crítica es la medida  $Y$  que corresponde a la menor distancia medida en ese intervalo, es decir, es la más cercana al robot dentro del grupo ya que este punto puede tomar más rápido contacto con el robot y detenerlo o desviarlo.

La división en regiones permite hallar los valles libres y los picos llenos de obstáculos.

Después de realizadas las tres tareas, finalmente este bloque entrega al siguiente dos vectores a manera de observación ( $O$  según la ecuación 3) que son:  **$O_1$**  el vector  $G$  de 36 posiciones que contiene la distancia crítica debido a los obstáculos allí presentes, y  **$O_2$**  que es un vector  $C$  con las características para la identificación de obstáculos.

### 3.3.2. PRE-PROCESAMIENTO CÁMARA

Por el lado de la cámara, este bloque recibe una imagen de tamaño 640x480x3 en RGB capturada a través del software Player para realizarle tareas de: reducción de tamaño, eliminación de ruido, cambio de espacio de color y transformación de la información.

Aplicando la ecuación 3 del capítulo 2,  $X = 2$  ya que es el segundo sensor que envía información al sistema de fusión,  $E$  = intensidad luminosa capturada en una imagen,  $Y$  es igual al valor en terna debido a que la cámara está en RGB. Al igual que en el sensor láser, la incertidumbre del dato ( $\Delta Y$ ) no se toma, en este caso se corrige en el bloque pre-procesamiento (ruido en imagen) y el tiempo ( $t$ ) tampoco se usa porque se recibe la última imagen capturada por la cámara cuando Player le solicita datos.

Al igual que otros sensores, la cámara no es inmune al ruido presente en las medidas que toma. La utilizada en este trabajo de grado (Logitech HD C270) tiene una resolución de video de 720p, lo que disminuye el ruido presente en las medidas. Aún así, se realiza una reducción de tamaño en la imagen en pro de: reducir el ruido remanente y disminuir la carga computacional en el análisis de la imagen obtenida a través de los siguientes procesos del sistema de fusión sensorial.

La imagen en RGB se reduce a  $320 \times 240 \times 3$  y luego se convierte a una imagen en YIQ para hacer ágil y rápido la aplicación de los métodos de visión artificial elegidos.

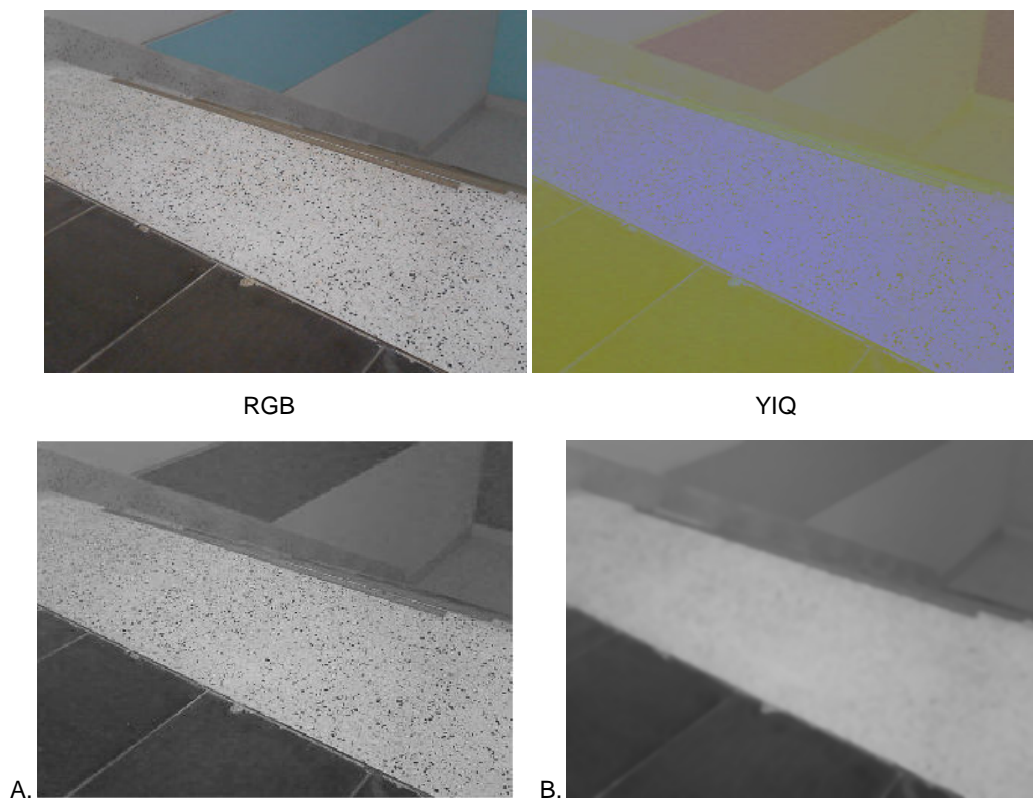
Después de la reducción, el cambio del espacio de color RGB al YIQ es realizado dado que la información de color e intensidad de luz está presente en tres lugares diferentes (matriz R, G, B) y un cambio en alguna de ellas afecta el total de la información, además se deben tener todas tres para analizar un mismo punto en la imagen. Además, se tiene la separación de la intensidad de luz contenida en la matriz Y obteniendo una imagen en escala de grises mientras que las matrices I, Q almacenan la información de color. Conviene entonces, seleccionar la matriz Y para los siguientes procesos ya que es necesaria para llevar a cabo la aplicación y conserva buena parte de la información.

La cámara Logitech HD C270 tiene una característica de corrección de luz (**Fluid Crystal** Tabla 3.2), a pesar de eso, se hace necesaria la aplicación de filtros a la imagen.

Primero se utiliza el filtro mediana para descartar algún valor pico aislado en el nivel de gris y obligar a que los pixeles tengan un valor semejante a sus vecinos.

Después se aplica el filtro gauss para disminuir las transiciones bruscas de nivel de gris entre pixeles adyacentes, facilitando su agrupación y posterior procesamiento en el siguiente bloque.

A continuación, se muestra una imagen de ejemplo para ilustrar el procedimiento en el bloque.



Componente Y, antes (A) y después (B) de la aplicación de los filtros.

Figura 3.7. Transformación de la imagen a través del bloque. [Fuente propia]

Por último, se entrega al siguiente bloque, una imagen de 320x240x1 en escala de grises usando la componente Y en forma de observación **O3**.

### 3.3.3. PROCESAMIENTO CÁMARA

Este bloque recibe la observación **O3** del bloque anterior y empieza el procesamiento con base en el algoritmo descrito en [Zhan, Huang, Wu 08]. Este bloque se encarga de dos tareas: una implícita que es hallar la continuidad del piso y otra explícita la cual es extraer características del obstáculo con mayor riesgo en el ambiente a navegar: las escaleras hacia abajo.

Se empieza generando una ROI\_1 en la parte inferior de la imagen ya que contiene la mayor probabilidad de obtener el nivel de gris para referencia del piso.

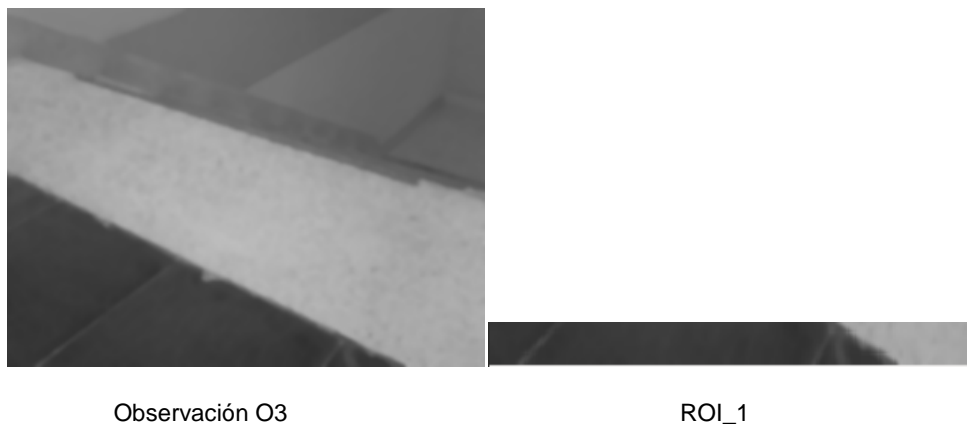


Figura 3.8. Preparación de la imagen antes de aplicar el método en componente Y. [Fuente propia]

Ciertamente, es necesario garantizar que el valor de referencia sea aquel que pueda caracterizar el piso y así hallar la continuidad del mismo en las imágenes siguientes. Con lo anterior, se hace preciso aplicar condiciones.

En [Zhan, Huang, Wu 08] se describen tres restricciones para la aplicación del método de navegación usando una cámara monocular, las cuales se acogen como condiciones para la posición inicial del robot (antes de iniciar su navegación):

- El piso en la posición inicial del robot es favorable, porque no contiene reflejos, brillos o grietas.
- El piso es uniforme y no es inclinado.
- No contiene obstáculos que cubran la imagen obtenida por la cámara.

Una vez aplicadas las restricciones, se busca ese nivel de gris referencia en la ROI\_1, y luego de hallado se crea un rango de detección (referencia  $\pm 15$ ) que sirve para buscar la continuidad del piso en la imagen **O3**.

Luego, se toma ese rango para calcular la proporción de piso, aplicándolo como un umbral a la imagen **O3** (figura 3.9).

Cuando la proporción del piso en comparación con el total de la imagen es menor al 30%, se procede a analizar **O3** para caracterizar las escaleras hacia abajo ya que no son detectados por el haz de láser (debido a su ubicación debajo del mismo).

En algunas ocasiones, el cambio en la continuidad del piso se debe a que la cámara captura una imagen de la pared, pero esta eventualidad queda superada porque el sensor láser mide la distancia y envía esa medida al sistema de fusión sensorial para que pueda tomar acciones evasivas.

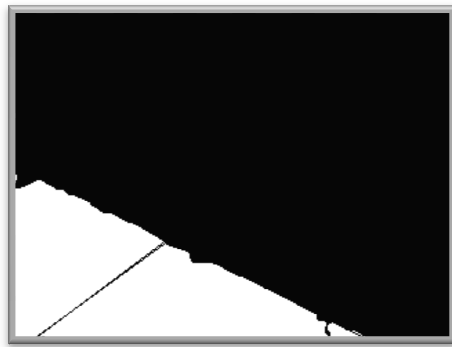


Figura 3.9. O3 después de aplicado el umbral. [Fuente propia]

Lo siguiente, la extracción de características de la escalera, se hace en base a [Shuihua, Yingli 12] que describe una forma de detectarlas usando una cámara que entrega imágenes en RGB, utilizando la transformada Hough.

La imagen **O3** se umbraliza para facilitar la aplicación de la transformada Hough. Después de obtenidos los valores de rho y theta, estos se comparan con una base de datos para saber si pertenecen o no a un escalón hacia abajo.

A continuación, se muestran imágenes durante la aplicación de la transformada.

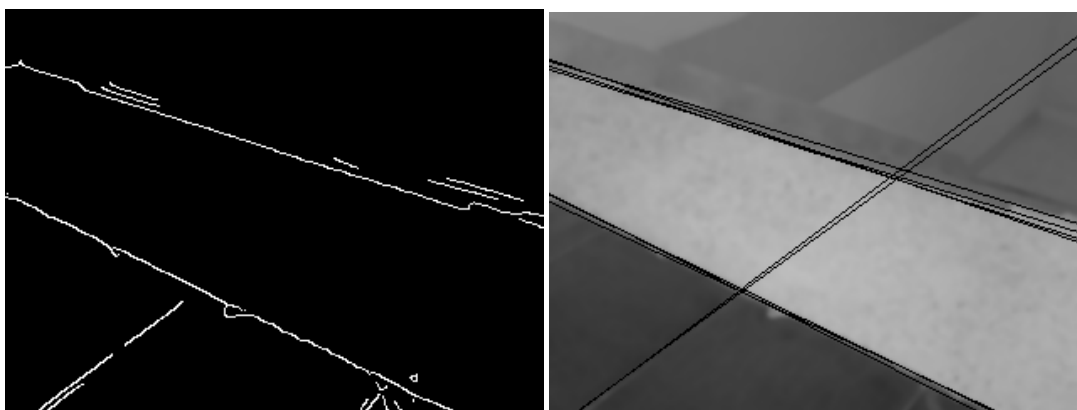


Figura 3.10. Extracción de características rho y theta. [Fuente propia]

Por último, se entrega al siguiente bloque la observación **O4** que contiene la proporción de líneas Hough detectadas dentro de la base de datos.

### 3.3.4. FUSIÓN SENSORIAL

Este bloque recibe las observaciones **O1**, **O2** y **O4**, con el objetivo de identificar los obstáculos en el FoV donde se cruzan tanto el sensor láser como la cámara, para entregar dos observaciones (**O5**, **O6**) que permitan modificar la estrategia de navegación del robot en el siguiente bloque.

Es importante mencionar, que para realizar la fusión sensorial entre el sensor láser (mide distancia con resolución mayor a milímetros) y una cámara monocular (imágenes en RGB), la información extraída de ambos debe tener el mismo formato y representar información similar.

Existen variadas técnicas de fusión sensorial que obligan a que la información tenga un formato igual a alguno de los sensores. Para este trabajo de grado, tendría la cámara que entregar medidas de profundidad o generar una imagen a partir de las medidas de distancia del láser.

Sin embargo, la técnica de la Red Neuronal no necesita cambiar a un formato de información que entrega alguno de los sensores, ya que recibe valores que representan características las cuales son independientes de cualquier formato con referencia a cualquier sensor.

En el caso del láser que mide distancias con resolución mayor a milímetros, se obtienen tres características y con la cámara se consigue una. La salida de la Red Neuronal son cuatro etiquetas que representan de manera general, los obstáculos presentes en el ambiente estructurado: Plano, Redondo, Desconocido y Escalera (Figura 3.11).

Para entender mejor las características y lo que representan, se construyen las condiciones que las generan a partir de **reglas de fusión**.

La creación de las características a partir de reglas de fusión se fundan en el trabajo **[Flynn 88]** (se considera canon) que describe esta técnica. Es elegida porque la información que usa para la decisión no exige mucho procesamiento y los procedimientos no son complejos. La formulación de esta regla de fusión se hizo siguiendo la metodología descrita en la sección 2.3.5.

Los valores que pueden tomarlas características son:

- 0.0, que significa error o ausencia.
- 0.2, representa una característica débil.
- 0.4, indica características de círculo.
- 0.8, tiende a ser plano.

Se utiliza un conjunto desde 0.0 hasta 0.8 para asignarlos a las características en pro de normalizar las cifras de cada proceso de extracción, ya que cada una tiene una escala diferente.



La primera característica utiliza la teoría geométrica de ángulo capaz descrita en [Xavier, Pacheco, Castro, Ruano, Nunes 04] en el método **Inscribed Angle Variance**. Después del procedimiento el resultado es un ángulo promedio ( $\beta$ ) entre  $90^\circ$  -  $180^\circ$  y una varianza (en grados) del grupo ( $\sigma_1$ ) que me permite obtener la característica de **ángulo de obstáculo**.

El valor de esta característica se asigna de acuerdo con la siguiente condición:

$$\begin{aligned}
 & \text{Si } \beta < 80 \text{ ó } \beta > 180 \quad C1 = 0.0 \\
 & \text{De lo contrario: } \beta + \sigma_1 - 180 = P \\
 & \quad \beta - \sigma_1 - 90 = Q \\
 & \text{Si } P < Q \text{ y } |P| < 21 \rightarrow C1 = 0.8 \\
 & \text{Si } Q < P \text{ y } |Q| < 36 \rightarrow C1 = 0.4 \\
 & \text{ninguna de las anteriores,} \quad C1 = 0.2
 \end{aligned}$$

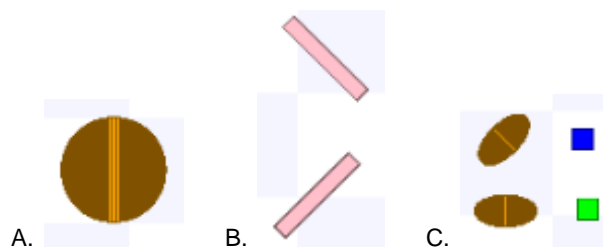


Figura 3.11. A) Redondo. B) Plano. C) Desconocido. [Fuente propia]

La segunda característica usa la ecuación geométrica de la pendiente. Es el resultado de promediar la pendiente calculada entre las medidas  $Y$  sucesivas después de realizar la descomposición vectorial de  $Y$  debido a que se cuenta con la distancia (magnitud) y el ángulo de medida (ángulo). Se obtiene la pendiente ( $m$ ) en milímetros y su varianza ( $\sigma_2$ ) para generar la característica **inclinación de obstáculo**.

Después de varias sesiones de pruebas, se observa que el valor de  $m$  y  $\sigma_2$  es muy pequeño, por tal motivo se multiplican por 1000. El valor de la inclinación se asigna de acuerdo con la siguiente condición:

$$\begin{aligned}
 & \text{Si } |m| < 100 \text{ y } \sigma_2 \leq 200 \rightarrow C2 = 0.8 \text{ (plano horizontal)} \\
 & \text{Si } |m| < 20000 \text{ y } \sigma_2 \leq 30000 \rightarrow C2 = 0.8 \text{ (plano vertical)} \\
 & \text{Si } 100 \leq |m| < 1000 \text{ y } 200 < \sigma_2 \leq 9000 \rightarrow C2 = 0.4 \text{ (redondo)} \\
 & \text{de lo contrario,} \rightarrow C2 = 0.2
 \end{aligned}$$

Es de anotar, que el plano horizontal y vertical, se determinan respecto a la orientación del robot: de frente al robot es horizontal y paralelo es vertical.

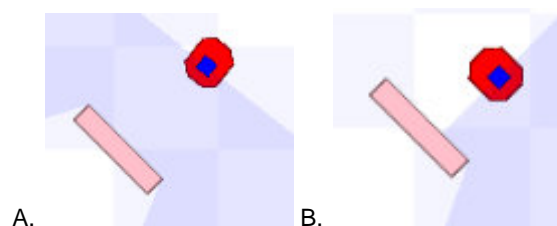


Figura 3.12. Determinación de plano horizontal (A) y vertical (B) respecto al robot. [Fuente propia]

La tercera característica se genera haciendo la descomposición vectorial del punto inicial y final para obtener una referencia a partir de las componentes en el eje Y eligiendo el menor valor. Luego, se compara cada componente en Y de los puntos dentro del grupo y se busca aquellos que sean menores a la referencia: entre mayor sea la proporción (en porcentaje) de puntos comparado con el total del grupo, hay un indicador de curvatura; si la proporción es mínima (casi cero) es un indicador de ausencia de curvatura. La característica de **curvatura de obstáculo** toma su valor de acuerdo a la siguiente condición:

*referencia > componente Y punto i del grupo*

*No → no agregar;*

*Sí → agregar;*

$$P = \left( \frac{\text{total suma}}{\text{total grupo}} \right) * 100$$

$$P \geq 80.0 \rightarrow C3 = 0.4$$

$$P \leq 25.0 \rightarrow C3 = 0.8$$

$$\text{de lo contrario, } C3 = 0.2$$

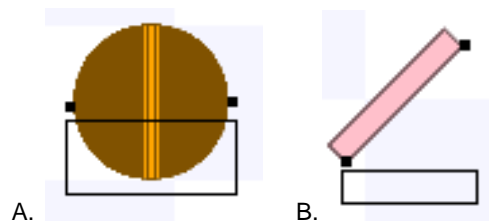


Figura 3.13. Los puntos indican los extremos y el rectángulo los puntos bajo la referencia. A) es para curvatura redonda mientras B) es para plana. [Fuente propia]

La cuarta característica se toma del bloque cámara aplicando la siguiente condición a la **O4**:

$$O4 \geq 30.0 \rightarrow C4 = 0.8$$

$$\text{de lo contrario, } C4 = 0.0$$

Una vez derivados los valores de las características, se ingresan a la RNA para la identificación del obstáculo. A continuación, se muestran los casos ideales en la clasificación por etiquetas.

CARACTERÍSTICAS DE ENTRADA				ETIQUETA DE SALIDA
C1	C2	C3	C4	
0.8	0.8	0.8	0	Plano
0.4	0.4	0.4	0	Redondo
0.2	0.2	0.2	0	Desconocido
---	---	---	0.8	Escalera

Tabla 3.3. Entradas y salida de la RNA.

Por cierto, del bloque procesamiento láser se reciben dos observaciones (**O1**, **O2**) y una del bloque cámara (**O4**). Para la identificación del obstáculo se usan **O2** y **O4** en la RNA. **O1** se emplea en otra tarea.

**O1** es un vector que contiene la distancia crítica, y esta se usa para complementar la identificación del obstáculo, dado que no es suficiente con etiquetar el obstáculo sin saber si puede colisionar o no con el robot y cambiar su trayectoria, detenerlo o causarle daño.

A partir de **O1** se establece el riesgo de colisión mediante la siguiente condición:

- $1.2 < \text{distancia crítica} \leq 8.0$ , lo que significa riesgo bajo (1) ya que la distancia de los posibles obstáculos no detiene ni disminuye el movimiento del robot.
- $0.5 < \text{distancia crítica} \leq 1.2$ , se asigna riesgo medio (2) debido a que los obstáculos pueden o no detener el robot dependiendo de la orientación y movimiento del mismo.
- Distancia crítica  $\leq 0.5$ , se asigna riesgo urgente (5) porque estos obstáculos detienen o disminuyen el movimiento del robot.

Aplicada la condición a la observación **O1**, se asigna a otro vector G el riesgo de colisión (**O6**).

Es necesario resaltar, que hay dos obstáculos con la mayor dificultad en la detección: las escaleras hacia abajo que están por fuera del haz del láser, y el vidrio cuya transparencia lo hace “desaparecer” en la imagen.

Aunque el vidrio no aparece en la imagen captada por la cámara puede ser detectado por el láser; y sucede igual para la escalera por debajo del rango del láser que sí se muestra en la imagen.

Ahora veamos cada caso por separado.

- Para el caso del vidrio.

Se analizan tres situaciones con el vidrio. El primer caso es el vidrio limpio, el segundo es de aspecto corrugado, y por último se tiene el vidrio cubierto con papel contact.

Respecto al primer caso, el vidrio no aparece en la imagen ya que es transparente y no afecta la luz que lo atraviesa. Su influencia en el cálculo de proporción de líneas Hough es insignificante porque no se agregan rayas debido a las tenues sombras y reflejos en su superficie.

A pesar de eso, su presencia representa riesgo para el robot y puede detenerlo o desviarlo con un choque. Por el lado del láser, el vidrio se detecta como un obstáculo con diferente geometría; puede ser plano o desconocido dependiendo del resultado en el proceso de extracción de características. Por consiguiente, se puede detectar y evadir (Figura 3.14).

En seguida, se muestra una imagen del vidrio al frente del robot, sostenido por dos objetos bajo el haz laser y también la salida del sensor laser leída a través del software Player. El recuadro rojo indica la zona que capta la cámara en el rango del sensor laser.

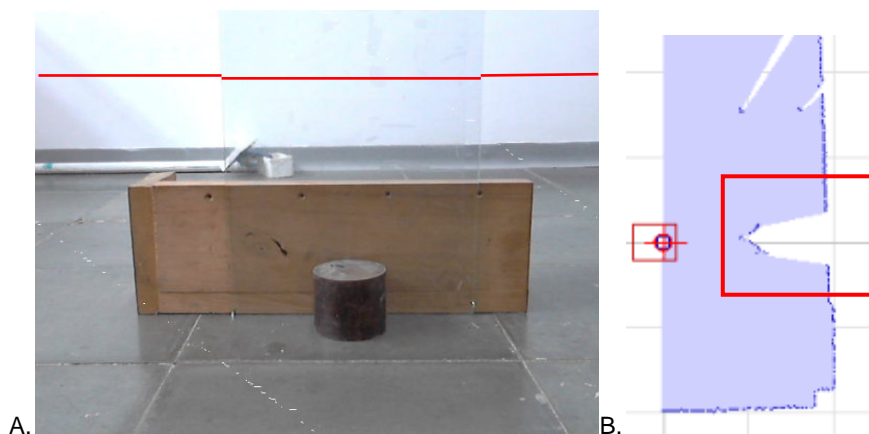


Figura 3.14. A) muestra la posición del vidrio y la línea roja indica el haz láser. B) contiene las medidas hechas por el láser mientras que el cuadro rojo indica la parte captada por la cámara. [Fuente propia]

- Caso del vidrio con aspecto corrugado.

El segundo caso muestra como resultado del bloque fusión sensorial la etiqueta de desconocido debido a las lecturas del sensor láser frente al vidrio. En la siguiente figura, se muestra un ejemplo de lo que sucede con las medidas realizadas por el sensor láser.

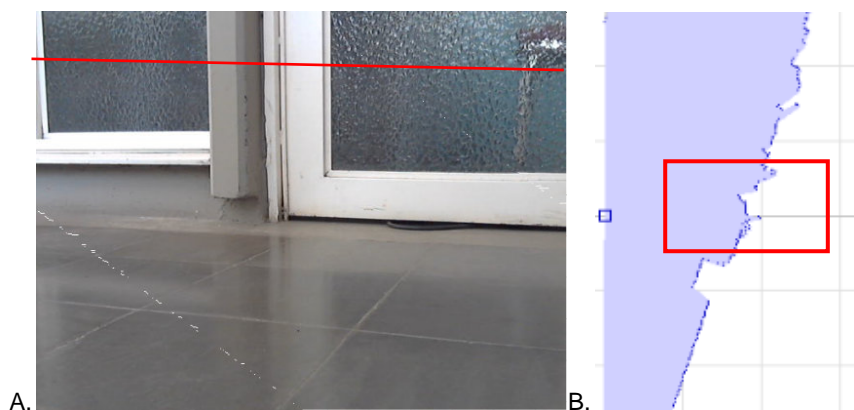


Figura 3.15. A) muestra el vidrio con aspecto corrugado. La línea roja indica el haz láser. B) contiene las medidas hechas por el láser mientras que el cuadro rojo indica la parte captada por la cámara. [Fuente propia]

Las medidas (recuadro rojo) parece que saltan y no tienen continuidad haciendo que la forma plana del vidrio se pierda. Lo anterior se debe a la forma de la superficie en el vidrio (aspecto corrugado) que distorsiona la medida que hace el sensor.

A pesar de eso, el vidrio aún se puede detectar, clasificar con alguna de las cuatro etiquetas y evadir.

- Caso del vidrio cubierto.

En los casos anteriores el vidrio no se encontraba con la superficie cubierta lo cual evitaba la identificación de la forma. Sin embargo, cuando se le coloca una envoltura

(sea papel contact, película de color) las medidas del sensor láser no se distorsionan y la forma plana aparece, además, la envoltura de color permite la aparición del vidrio en la imagen.

Un ejemplo de ello, es la siguiente figura.

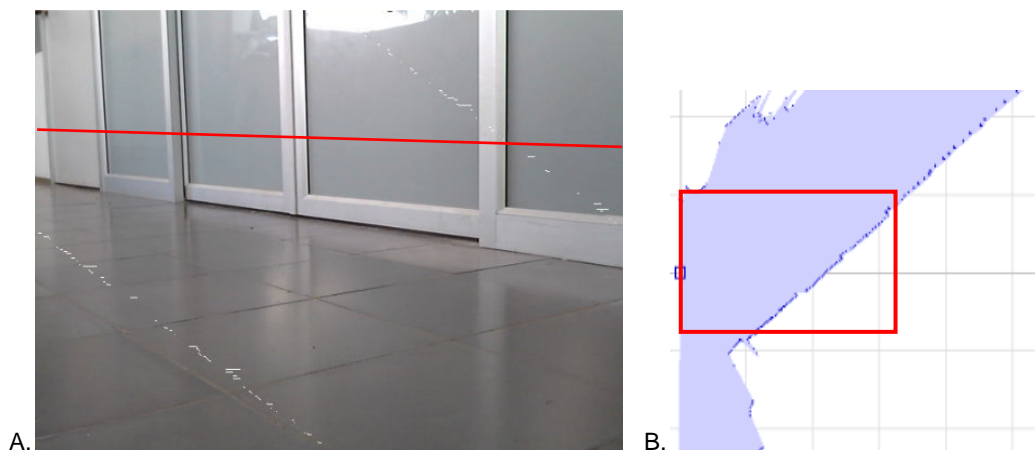


Figura 3.16. A) muestra el vidrio cubierto, la línea roja indica el haz láser. B) contiene las medidas hechas por el láser mientras que el cuadro rojo indica la parte captada por la cámara. [Fuente propia]

En el recuadro rojo se ven las medidas del láser que corresponden al área que la cámara capta en la imagen. Aquí se muestra la continuidad de las medidas y la ausencia de distorsión. También que el vidrio cubierto aparece en la imagen, es de anotar que el color de la envoltura es similar al de las paredes del ambiente estructurado. Las medidas del sensor láser contra la pared y contra el vidrio cubierto son parecidas (Figura 3.17).

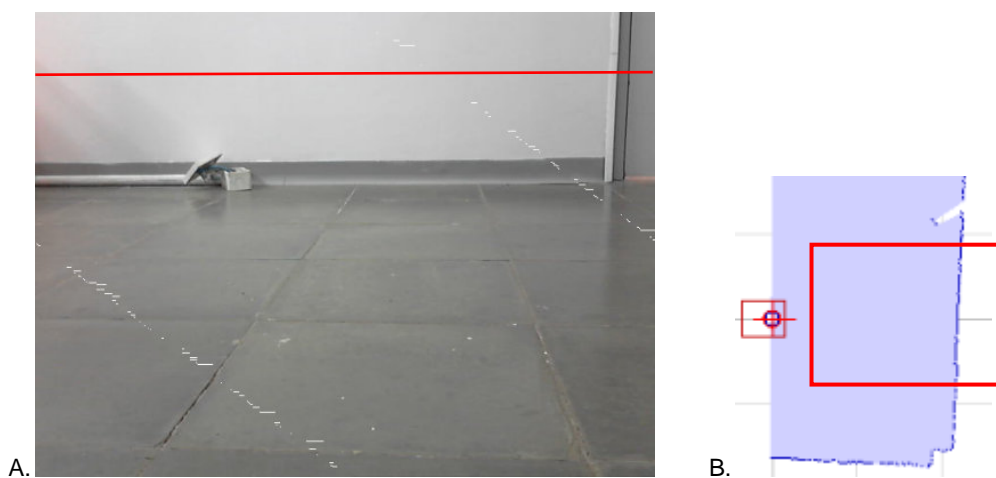


Figura 3.17. A) imagen de una pared. La línea roja indica el haz láser. B) contiene las medidas hechas por el láser mientras que el cuadro rojo indica la parte captada por la cámara. [Fuente propia]

- Para el caso de escaleras hacia abajo.

Están por debajo del haz láser por ende, no se pueden detectar, sin embargo, sí aparecen en la imagen. La detección de este obstáculo se asigna enteramente al bloque cámara ya que el sensor laser no puede detectarla.

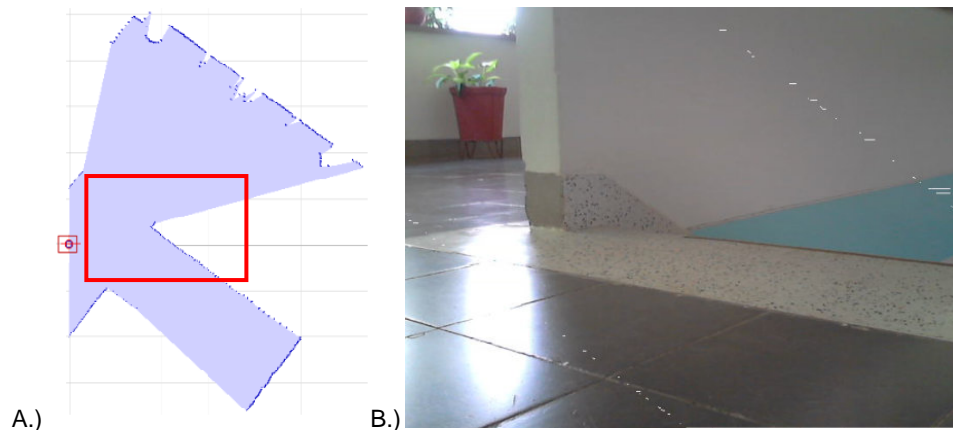


Figura 3.18. A) Rango del sensor láser al frente de las escaleras. B) Imagen tomada con el robot en el mismo punto que A. El cuadro rojo indica la zona que capta la cámara.

El peligro con las escaleras hacia abajo es que el laser le indica por distancia que no hay riesgo (le asigna riesgo 2 ó 1) ya que el sensor mide la distancia hacia la pared del fondo indicando una ruta segura; entonces concierne al bloque cámara enviar la característica correspondiente para detectar y evadir las escaleras puesto que el robot puede rodar en caída.

Hay una condición para evaluar antes de ejecutar la identificación del obstáculo y esta es:

- Distancia del obstáculo más cercano.

Esta condición evita que el algoritmo este ejecutándose sin necesidad, ya que no es necesario evadir un obstáculo si se encuentra a una distancia mayor o igual a 2 metros, ni es imperioso identificarlo.

Cuando aparece un obstáculo dentro del rango donde se realiza la extracción de características, se evalúa cuál es su distancia crítica, si es menor o igual a 2 metros, comienza el proceso de identificación usando las características obtenidas.

Indicado lo anterior, finalmente se entrega al siguiente bloque la observación **O5** que contiene la identificación del obstáculo y **O6** con el riesgo en cada posible ruta.

### 3.3.5. DEFINICIÓN DE TRAYECTORIA

Una vez recibidos los riesgos de colisión contenidos en **O6**, se compara cuál es la región que tiene el riesgo 5 (riesgo urgente) para evadirlo, puesto que no importa cuál obstáculo sea, la probabilidad de choque con el robot es alta.

Cuando se detecta un pico de obstáculo (riesgo 5) lo que se hace es evadirlo buscando los valles libres (riesgo 1, 2) que estén concatenados haciendo un valle más grande. En este caso el robot trata de evadir ese pico rápido, girando mientras se aleja del obstáculo.

La estrategia de navegación usada en este trabajo de grado es **wandering**, haciendo que el robot se mueva de frente hasta que aparece el riesgo 5 o la RNA del bloque anterior identifica el obstáculo, información contenida en **O6** y **O5** respectivamente.

Con base en **O5**, se modifica la forma de moverse del robot dentro del ambiente, y el cambio en la estrategia de navegación se hace en función los siguientes criterios:

- La distancia hacia el obstáculo.
- La etiqueta de identificación del obstáculo.
- Evaluar si ha terminado de evadirlo.

Después de revisar **O6** y verificar que no contiene picos de obstáculos, se evalúa la distancia del obstáculo más cercano al frente del robot (similar al bloque fusión sensorial) teniendo dos posibilidades:

- Si la distancia es mayor que 70 cm, el robot sigue de frente dado que no es necesario evadir un obstáculo que se encuentra alejado del robot.
- Si la distancia es menor o igual que 70 cm, se procede a analizar la información de **O5**.

El ángulo de dirección y la velocidad se transforman en un comando de dirección angular y otro de velocidad lineal usando el software Player, el cual permite controlar el giro del robot y la velocidad de los motores para la navegación.

En la siguiente tabla se muestran los movimientos del robot para evadir los diferentes tipos de obstáculos.

ETIQUETA	OBSERVACIÓN	MOVIMIENTOS
<b>PLANO</b>	Los obstáculos planos son las puertas, paredes, vidrios.	<ul style="list-style-type: none"> <li>• Distancia <math>\leq 70</math> cm.</li> <li>• Calcula la inclinación del obstáculo para alinearse con él.</li> <li>• Se alinea.</li> <li>• Después de evaluar que la distancia cercana al obstáculo sea mayor de 1 m., se aleja 30 cm del obstáculo para terminar de evadirlo.</li> <li>• Gira según el ángulo calculado anteriormente para volver a estar de frente.</li> <li>• Reanudar marcha hacia adelante.</li> </ul>

<b>REDONDO</b>	Se refiere a materas, canecas para residuos sólidos.	<ul style="list-style-type: none"> <li>• Distancia <math>\leq 70</math> cm.</li> <li>• Calcula el radio del obstáculo para describir un arco mientras lo evade.</li> <li>• Gira hacia la izquierda con un arco mucho menor, para alinearse con el obstáculo y evadirlo recorriendo el arco calculado antes.</li> <li>• Si ya ha transitado todo el arco, vuelve a quedar de frente. De lo contrario, sigue el recorrido.</li> <li>• Reanudar marcha hacia adelante.</li> </ul>
<b>DESCONOCIDO</b>	Piernas de personas, vidrios, o cualquier objeto que no sea plano o redondo.	<ul style="list-style-type: none"> <li>• Distancia <math>\leq 70</math> cm.</li> <li>• Calcula el radio del obstáculo para describir un arco mientras lo evade.</li> <li>• Se detiene y gira <math>90^\circ</math> hacia la izquierda para evadir el obstáculo recorriendo el arco calculado antes.</li> <li>• Después de transitar por el arco, vuelve a quedar de frente.</li> <li>• Reanudar marcha hacia adelante.</li> </ul>
<b>ESCALERA</b>	Se detectan las escaleras hacia abajo.	<ul style="list-style-type: none"> <li>• Es detectada.</li> <li>• Detiene el robot.</li> <li>• Gira <math>180^\circ</math>, para evadir la escalera.</li> <li>• Detiene el robot, queda de frente ante una ruta libre de las escaleras.</li> <li>• Reanudar marcha hacia adelante.</li> </ul>

Tabla 3.4. Estrategia de navegación según la identificación del obstáculo. [Fuente propia]

Para una mejor comprensión de lo que el algoritmo hace, se coloca una ventana de observación para el usuario que muestra los riesgos por colores (1 = verde, 2 = amarillo, 5 = rojo) a partir de **O6** y también muestra la etiqueta de la RNA: Plano, Redondo, Desconocido, Escalera. La orientación del robot se señala con un cuadro azul dentro de las regiones del laser cuando modifica su normal trayectoria (ir hacia adelante). Ese cuadro indica el frente del robot y hacia qué dirección se mueve.

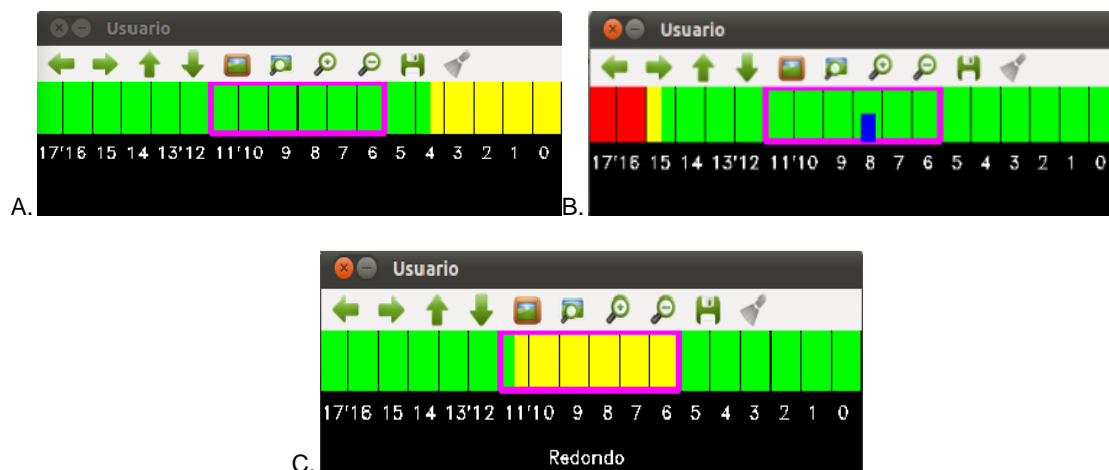


Figura 3.19. A) Visual para el usuario. B) El cuadro azul indica cambio de trayectoria. C) muestra la etiqueta de salida. El recuadro magenta es la zona donde se cruzan el FoV de cámara y láser. [Fuente propia]

Con la información en **O5** y **O6** se logra identificar los obstáculos detectados. Aunque esta clasificación es grupal basado en la geometría de los obstáculos y el nivel de gris, es suficiente para detectar distintos objetos que el robot se puede encontrar mientras hace navegación **wandering**.



Por último, se describe el software y programas usados para el funcionamiento de los bloques de la figura 3.1.

### 3.4.1. PLAYER

Player 3.0.2 es una compilación de librerías que permiten manejar diferentes plataformas robóticas a través de interfaces comunes facilitando la reutilización de software y fortaleciendo las implementaciones multi-robot.

Player se comporta como un servidor albergando los dispositivos que se conectan con los sensores permitiendo el envío de comandos y recepción de datos entre los dispositivos y otras partes de Player como las librerías cliente o las aplicaciones (Figura 3.20).

Las librerías cliente son las que permiten programar en lenguaje C, C++ el código correspondiente a los nodos de fusión los cuales se encargan de las tareas de los bloques de la figura 3.1.

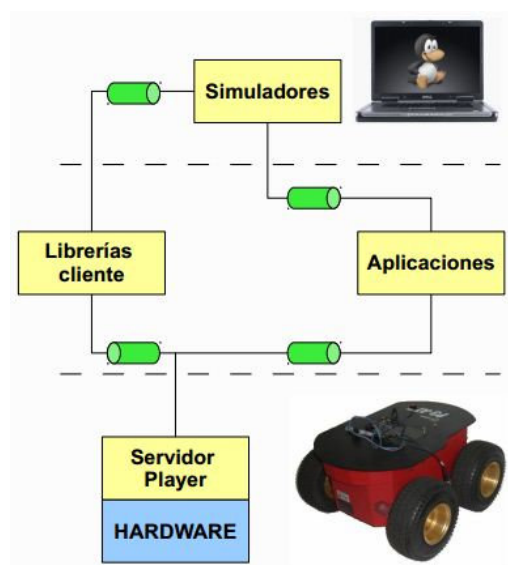


Figura 3.20. Interacción del proyecto Player con el robot. [Fuente Vaughan, Gerkey 06]

La figura 3.20 muestra la comunicación de todas las partes del proyecto Player con la plataforma robótica, en esta figura es el robot PioneerATX. Se debe anotar que las separaciones se deben a que las librerías cliente y las aplicaciones pueden interactuar con el servidor Player para usar al robot, o se simulan en un PC usando un complemento.

La comunicación entre el servidor Player y el robot se hace a través de una conexión de internet, ya que Player implementa el protocolo TCP/IP. El comando **player** inicia la recepción de datos con el robot y el envío de comandos para su movimiento. Aunque depende si el proyecto fue instalado vía descarga del archivo y posterior compilación (comando **player**), o vía gestor de paquetes Synaptic (comando **robot-player**).

Se puede obtener más información en las referencias [Vaughan, Gerkey 06], [Whitbrook 10].

### 3.4.2. STAGE

Es uno de los complementos de Player que permite la simulación de una plataforma robótica, diferentes sensores, distintos ambientes y/o varios robots a la misma vez.

El complemento Stage permite simular la cinemática de los robots, ambientes cerrados **indoor**, configuración de sensores y actuadores del robot, pero genera una vista en 2D del ambiente.

Stage puede generar una vista 2.5D proyectando la altura del ambiente, sin ser una vista 3D real. La figura 3.21 es un ejemplo de un ambiente simulado.

El otro complemento es Gazebo, que también es simulador. Gazebo permite simular la dinámica y cinemática en robots, ambientes cerrados al interior de edificios (**indoor**) y exteriores (**outdoor**), permite la configuración de los sensores y actuadores del robot y tiene una vista en 3D del ambiente y el robot simulado.

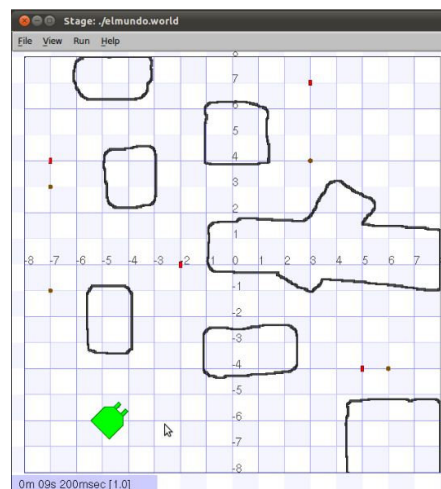


Figura 3.21. Simulación de un ambiente *indoor* con Stage. [Fuente: Owen 10]

Se elige el simulador Stage 3.2.2 porque provee un ambiente de pruebas simulado sin que sea necesario físicamente al robot, permite crear objetos en el ambiente, configurar el sensor láser (ya que Stage emula los dispositivos con buena fidelidad), y evaluar el código escrito.

Además se acopla bien con Player 3.0.2 y la versión 3.2.2 es la más rápida en ejecutarse comparada con las otras versiones de Stage. No carga el computador con gráficos cuyo procesamiento pueden afectar la velocidad de ejecución de la aplicación player (o robot-player) y/o el código cliente como la vista en 3D de Gazebo.

Las referencias [Whitbrook 10], [Stage 4] y [Owen 10] amplían el uso de este simulador.

### 3.4.3. OPENCV

OpenCV es uno de los proyectos que se pueden encontrar en la referencia [**@Willowgarage 5**], el cual consiste en una compilación de librerías que permiten el procesamiento rápido de imágenes, video y otras herramientas relacionadas con las imágenes.

Para el procesamiento de las imágenes captadas por la cámara, se usa OpenCV 2.4.0, aunque su última versión es 2.4.5; el último número indica cambios micro que se refiere a cambios en la sintaxis de algún comando, adición de un comando en otro lenguaje de programación, pero no hay cambios significativos en las librerías núcleo.

Se elige esta versión ya que permite la programación en lenguaje C, C++, Python además de contar con actualizaciones de funciones respecto a las versiones anteriores (la primera es 2.1.0), existen foros de ayuda y es una versión estable.

OpenCV significa Open source Computer Vision library que contiene muchas librerías para el procesamiento de imágenes [core, imgproc, video], ventanas [highgui], calibración de cámaras [calib3d], detección de características [features2d, objdetect], maquinas de aprendizaje [ml], clustering [flann, ml], aceleradores gráficos [gpu, ocl].

Para estudiar todos los alcances que tiene OpenCV, la referencia [**@OpencvDocumentation 1**] contiene la documentación y [**Bradski, Kaehler 08**], [**Laganière 11**] son libros guía.

### 3.5. COMUNICACIÓN PC – ROBOT

El robot recibe los comandos del bloque definición de trayectoria. Aunque es en el robot donde se ubican los sensores cámara y láser, en él no se ejecuta ninguno de los bloques de la figura 3.1. Estos bloques se realizan en un PC que se conecta al robot vía inalámbrica.

Para esa conexión, se debe tomar en cuenta que el Sistema Operativo del PC indica el protocolo a utilizar como puente entre la recepción de datos de los sensores a los bloques de pre-procesamiento y el envío de comandos al robot. Se elige instalar el Sistema Operativo Linux 11.04 Natty Narwhal porque:

- 1.) Aunque existen programas de robótica en Windows, los de Linux son gratuitos, tienen comunidades que brindan códigos de ejemplo y ayuda en la corrección de errores, variedad de plataformas robóticas, sensores, libre de licencias o compra de dispositivos en su mayoría.
- 2.) Tanto en Linux como en Windows, permite la simulación de un robot sin tener la necesidad de contar con él físicamente. Pero en los programas de Linux se puede contar con más dispositivos respecto a los de Windows.
- 3.) Al utilizar una maquina virtual que emule el S.O. Linux en Windows, divide los recursos físicos entre los dos S.O.

Como la comunicación se hace vía inalámbrica, se necesita también un protocolo de comunicación que permita el reconocimiento entre el PC y el robot: el protocolo ssh (Secure SHell), que se descarga usando Synaptic.

El protocolo ssh usa los siguientes paquetes para una óptima comunicación:

- Paquete ssh: Es un meta-paquete para instalar tanto el cliente como el servidor OpenSSH.
- Paquete openssh-client: Es una versión adaptable, ya que es una implementación libre del protocolo "Secure Shell" como cliente.
- Paquete openssh-server: Es una versión adaptable, ya que es una implementación libre del protocolo "Secure Shell" como servidor.
- Paquete ssh-import-id: Hace segura la conexión entre PC-robot mediante publickey.
- Paquete ssh-askpass-gnome: Usa de forma implícita el paquete anterior para interrogar por la clave del PC a bordo del robot y establecer la conexión.
- Paquete x11-session-utils: Contiene los paquetes necesarios para manejar una sesión remota, asignando el formato a los packet.

### 3.6. CONCLUSIONES

- El software Player ofrece muchas ventajas en cuanto a la comunicación con el robot sin ser necesaria la creación de un módulo que se encargue de esta.
- OpenCV es una librería que tiene muy buena velocidad de procesamiento de las imágenes, y está escrita en lenguaje C/C++ al igual que Player.
- Stage es un simulador que se complementa con Player y permite realizar pruebas y ensayos de los códigos clientes en el robot en distintos ambientes indoor. El simulador Stage permite realizar pruebas sin necesidad de la presencia física del robot.
- Mediante las sesiones de toma de datos, se observó que las distancias del láser facilitan la extracción de características de los obstáculos y la detección de regiones usando las discontinuidades en las medidas de distancia facilitando la segmentación de las medidas del sensor laser y entregar al sistema de fusión la información necesaria para la extracción de características.
- La fusión de datos se puede realizar siempre que la información tengan alguna similitud que permita su comparación o contraste. Para este trabajo de grado,

se utiliza la técnica de RNA que permite la clasificación de obstáculos mediante características. La RNA es independiente del formato de información de los sensores ya que recibe como patrón de entrada un vector con valores.

- La RNA tiene la capacidad de generalizar los patrones de entrada, eso permite reducir la cantidad de datos de entrenamiento para evitar colocar todos los casos posibles.
- La implementación de los nodos de fusión debe ser cuidadosa para evitar errores de programación que afecten el desempeño del sistema y caiga equivocadamente dentro de la fusión catastrófica.
- Aunque algunos de los criterios para evaluar la fusión sensorial no se pueden medir, se puede ver su efecto en las sesiones de pruebas y ensayos buscando cuál es la característica que lo causa mediante una revisión desde el final al inicio y desde el comienzo al final a lo largo del sistema de fusión implementado.

## 4. PRUEBAS Y RESULTADOS

### 4.1. INTRODUCCIÓN

En este capítulo se presentan las pruebas para evaluar la fusión sensorial desarrollada confrontada con un algoritmo de navegación canon usando el láser (VFH) y con otro que emplea el criterio más utilizado en la navegación usando cámaras: la ubicación del obstáculo.

Se realizan cuatro pruebas: en primer lugar, se tienen las pruebas que miden el desempeño de la navegación utilizando los sensores por separado (una para el laser y otra para la cámara). Después, se presentan dos pruebas de las cuales se puede obtener el desempeño de la fusión sensorial en dos aspectos: identificación y cambio en la estrategia de navegación.

En la sección de resultados y análisis, primero se muestra el escenario de prueba mediante una figura con la posición inicial del robot (círculo rojo) y otras posiciones donde ocurren eventualidades en el algoritmo evaluado, luego se reporta una tabla con los resultados obtenidos y algunas observaciones encontradas. Después se hace un análisis general de la prueba.

### 4.2. DESCRIPCIÓN DE LOS ESCENARIOS DE PRUEBA

Antes de detallar las pruebas, es conveniente describir los escenarios donde se realizan y los obstáculos contenidos allí.

El laboratorio de robótica móvil es un recinto cerrado que contiene una mesa de 2.4m de ancho y 75 cm de alto, otra mesa cubierta por un mantel negro de 1 m de ancho por 83 cm de alto. La puerta tiene 90 cm de ancho y los conos plásticos usados tienen una base cuadrada de 18 cm x 18 cm con una altura de 45 cm.

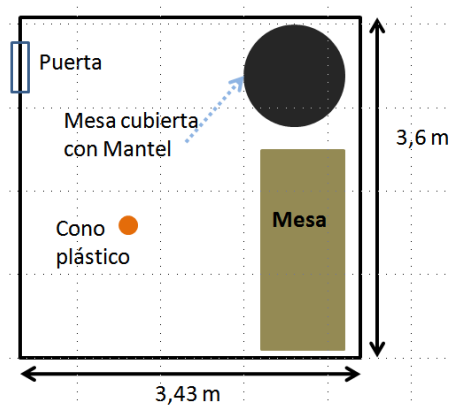


Figura 4.1. Laboratorio de Robótica móvil. Muestra las dimensiones del lugar y los obstáculos que contiene. [Fuente propia]

El segundo piso de la Escuela de Ingeniería Eléctrica y Electrónica se compone de pasillos de distinta amplitud, puertas de madera y vidrio cubierto, paredes, escaleras hacia abajo, materas y una caneca para residuos sólidos. Las medidas de las puertas son: P con ancho de 80 cm, P1 mide 180 cm y P2 90 cm. Las materas son los círculos de color naranja y tienen medidas entre 47 – 40 cm de ancho y 32 – 50 cm de alto. La caneca (círculo azul) mide 47 cm de ancho y 80 cm de alto. La entrada hacia las escaleras es de 1.54 m de ancho. La altura de las paredes y de las puertas no se toma ya que superan ampliamente la ubicación del haz láser en el robot. La línea azul en la parte inferior indica que ese lado contiene vidrio cubierto con una película opaca.

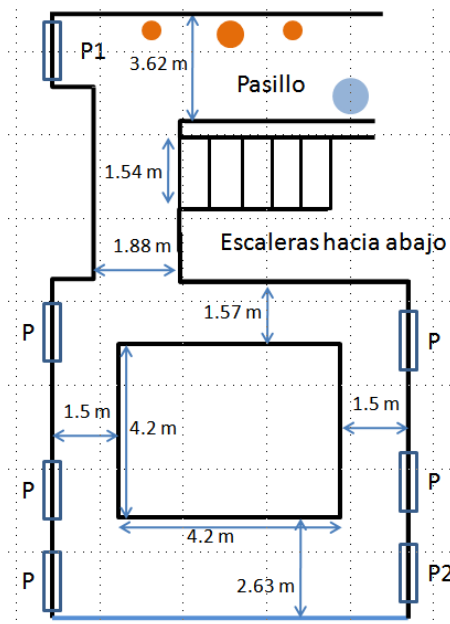


Figura 4.2. Segundo piso de la Escuela de Ingeniería Eléctrica y Electrónica. Contiene las medidas de los pasillos y la ubicación de los obstáculos. [Fuente propia]

### 4.3. PRUEBA DE NAVEGACIÓN CON EL SENSOR LÁSER.

#### 4.3.1. Descripción

En esta prueba se implementa el algoritmo VFH descrito en [Borenstein, Koren 91] utilizando solamente el sensor láser, para evaluar la detección y evasión de obstáculos en los escenarios de prueba. Se realizó tres pruebas, cada una en horarios diferentes: mañana (8 am), mediodía y tarde (4 pm).

#### 4.3.2. Laboratorio de Robótica móvil. Resultados y análisis.

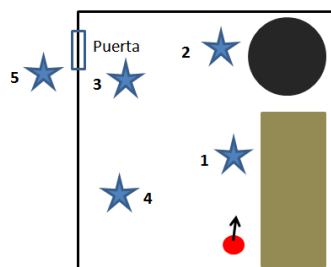


Figura 4.3. El círculo rojo indica la posición inicial del robot en el ambiente y la flecha su orientación. Las estrellas son posiciones donde ocurren cambios en la navegación. [Fuente propia]

Aspecto	Porcentaje de acierto	Observaciones
Detección de obstáculos	100%, 0%	Detecta obstáculos con base en la distancia hacia el robot (100%). Aunque puede detectar las patas de la mesa, no es así para la tabla de esta (0%). En general, se relaciona con el literal A, C.
Evasión de obstáculos	100%, 50%	El 100% se relaciona con el literal C. Para el caso de las mesas el porcentaje es 100% con la marca y 0% sin la marca; de ahí que sea 50% porque se recurre a una marca externa, como se indica en el literal A.
Navegación del robot	---	Movimiento continuo y sin cambio abruptos. También puede hallar la salida de la habitación si está abierta como se indica en el literal B
Evitar riesgo para el robot	100%, 0%	El robot evita los obstáculos alejándose de ellos como se describe en el literal C indicando 100%. Pero con las mesas, es necesaria la marca para evitar un choque como se muestra en el literal A. Con la marca, detecta y evade (100%), sin ella, hay choque (0%).

Tabla 4.1. Resultados en el escenario 1.

- A. **Las mesas son obstáculos difíciles de evadir.** Al encontrarse con mesas (figura 4.3), detecta las patas de estas pero se mueve hacia el espacio debajo de ella. Cuando se presenta este caso en la posición 1, se detiene el robot y se examina lo que hace el algoritmo mostrando un valle libre entre las dos patas de mesa; pero no se debe tomar esa ruta porque el robot cuenta con una estructura que aumenta su altura, haciendo que se estrelle con la mesa y algunas de estas tienen varillas entre las patas de la mesa (figura 4.4). Para marcar ese espacio como ruta no segura, se coloca un cono plástico el cual es detectado y evadido por el robot. Con la marca detecta y evade el espacio bajo la mesa (100%); sin la marca no (0%).

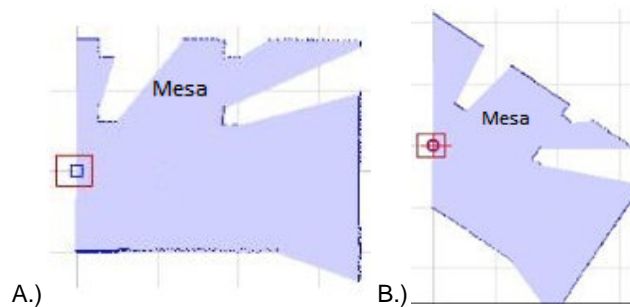


Figura 4.4.A.) Medidas del láser cuando el robot pasa al lado de una mesa alta. B.) Valle libre bajo la mesa, se registra un falso positivo. [Fuente propia]

- B. **Puede hallar la salida si la puerta del Laboratorio está abierta.** El algoritmo VFH busca el valle libre más cercano al punto de llegada. En la figura 4.3 cuando la puerta está cerrada, el robot llega hasta la posición 3 y luego va hacia 4; si la puerta está abierta, el robot llega al punto 5 saliendo del laboratorio. En ambas pruebas el punto de llegada se reemplaza por una dirección deseada (seguir al frente) y el algoritmo puede hallar la salida si esta se encuentra habilitada y sin obstáculos.
- C. **Detecta y evade obstáculos.** Las paredes, la puerta y el mantel (posición 2 figura 4.3) son detectados y evadidos.



### 4.3.3. Segundo piso Escuela de Ingeniería Eléctrica y Electrónica. Resultados y análisis.

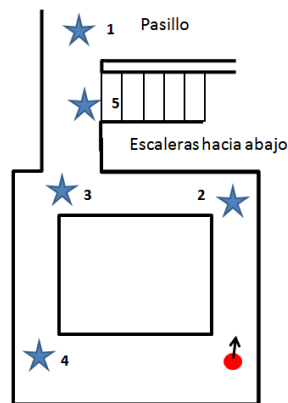


Figura 4.5. El círculo rojo indica la posición inicial del robot en el ambiente y la flecha su orientación. Las estrellas son posiciones donde ocurren cambios en la navegación.

Aspecto	Porcentaje de acierto	Observaciones
Detección de obstáculos	100%, 0%	Detecta obstáculos con base en la distancia hacia el robot (100%). En general, se relaciona con el literal D, E. La escalera hacia abajo está debajo del haz láser, por tanto, su detección es 0% como se amplía en el literal F.
Evasión de obstáculos	100%, 0%	El 100% se relaciona con el literal D, E. Con escaleras el valor es 0% referido en el literal F.
Navegación del robot	---	Movimiento continuo y sin cambio abruptos, como indica en los literales D, E, G.
Evitar riesgo para el robot	100%, 0%	El robot evita los obstáculos alejándose de ellos como se describe en el literal E indicando 100%. El 0% es para la escalera porque no se puede detectar ni evadir como se explica en el literal F.

Tabla 4.2. Resultados en el escenario 2.

**D. El robot se mueve sin hacer oscilaciones.** El robot se mueve sin cambios abruptos de dirección entre las paredes de los pasillos sin hacer oscilaciones bruscas (figura 4.6). Sucede así cuando va desde la posición inicial hasta 2 y luego hacia 3 en la figura 4.5. En el momento que se encuentra en un amplio corredor, sigue adelante hasta que el riesgo aumente indicando la presencia de un obstáculo, empezando a buscar el valle libre más amplio para dirigirse allí como sucede en la posición 1 y 4 de la figura 4.5.

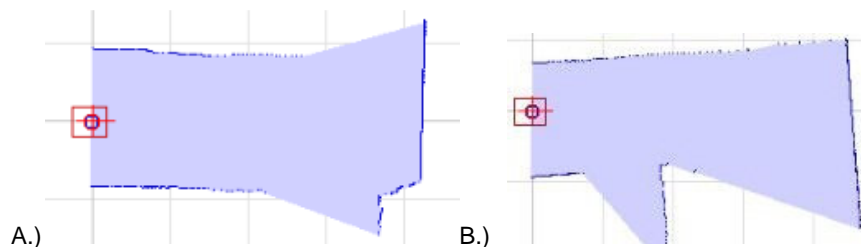


Figura 4.6. A.) Medidas del láser cuando se encuentra en el pasillo desde la posición 2 hacia la 3. B.) El robot navegando desde la posición 3 hacia la 5. [Fuente propia]

- E. **Detecta y evade obstáculos.** La detección funciona pero la evasión varía de acuerdo al obstáculo. En cuanto a los estáticos como paredes, puertas, canecas para residuos el robot los detecta y se mueve en la dirección contraria a la ubicación del obstáculo buscando alejarse de él. Para los dinámicos como las personas que se mueven cerca del robot, aunque las detecta, no es capaz de predecir la trayectoria de movimiento de la persona y algunas veces falla al tratar de evadirlas ya que se mueve en la misma dirección de las personas, otras se le cruza por delante y algunas vacila girando antes de elegir un valle libre como ruta segura.
- F. **Cuando se encuentra al frente de escalones hacia abajo, el robot sigue de frente.** En la posición 5 (figura 4.5) se examina lo que el algoritmo hace y muestra un valle libre haciendo que el robot siga de frente, pero esa ruta es engañosa porque NO ES UNA RUTA SEGURA (figura 4.7).



Figura 4.7. Medidas del láser cuando el robot está al frente de la escalera. Se ve una zona libre registrando un falso positivo. [Fuente propia]

- G. **El horario de las pruebas no es determinante.** El horario solamente influye en la concurrencia de personas, siendo la mañana con mayor afluencia, el mediodía con menos y la tarde con la mínima cantidad, todas las personas son detectadas y evadidas.

En ambos escenarios, la prueba de navegación se hizo cinco veces en cada horario. Se debe anotar que es indiferente el tipo de obstáculo detectado (pared, puerta, pata de mesa, cono plástico, personas caminando) ya que el robot hace la misma acción de evasión: un giro sin detenerse ni realizarlo de manera abrupta hacia el lado contrario de la ubicación del obstáculo buscando alejarse de él.

#### 4.4. PRUEBA DE NAVEGACIÓN CON LA CÁMARA

##### 4.4.1. Descripción

En esta prueba se implementa el algoritmo descrito en [Zhan, Huan, Wu 08] el cual usa una cámara monocular para la navegación de un robot móvil mientras se evalúa la detección y evasión de obstáculos con este sensor, también el impacto del control en las fuentes de luz, ya que se utilizó una cámara. Se realizó tres pruebas cada una en horarios diferentes: mañana (8 am), mediodía y tarde (4 pm).

#### 4.4.2. Laboratorio de Robótica móvil. Resultados y análisis.

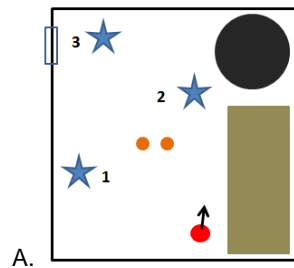


Figura 4.8. El círculo rojo indica la posición inicial del robot en el ambiente y la flecha su orientación. Las estrellas son posiciones donde ocurren cambios en la navegación. [Fuente propia]

Aspecto	Porcentaje de acierto	Observaciones
Detección de obstáculos	100%, 0%	Cuando el color del obstáculo es parecido al piso su detección es 0%, pero cuando es distinto es 100%. Aunque detecta obstáculos con base en la ubicación de estos en la imagen captada por la cámara (literal D), a veces falla (literal E).
Evasión de obstáculos	100%, 50%, 0%	Siempre que el obstáculo se pueda distinguir por su diferente color respecto al piso, la evasión es exitosa (100%). En otros casos (50%, 0%), depende de factores externos al robot, esto se amplía en el literal A, D, E.
Navegación del robot	---	Cuando se tiene la iluminación controlada, sucede lo descrito en los literales A, B, C. Algunas excepciones ocurren como se indica en el literal E.
Evitar riesgo para el robot	100%, 0%	Aunque puede detectar y evadir obstáculos con base en color de ellos (100%), a veces tiene errores (0%) que llevan al robot a un choque directo (literal E).

Tabla 4.3. Resultados en el escenario 1.

- A. **El control en las fuentes de luz influye de manera positiva en el algoritmo.** Cuando se controla la iluminación en el escenario de prueba, la navegación es más predecible: El robot empieza en la posición inicial y llega hasta la posición 3 de la figura 4.8.
- B. **El movimiento del robot es continuo y sin cambios abruptos.** El robot empieza su movimiento a través del lugar sin obstáculos presentes en el recorrido. Se tiene una continuidad en el movimiento mientras se mueve en la ruta elegida por el algoritmo.
- C. **Es posible determinar un trayecto de navegación.** Luego de varias pruebas, se halla una ruta predecible para el robot en este escenario y se colocan obstáculos como conos rojos para que sean detectados y evadidos. Se obtiene la detección y evasión, aunque el robot no termina en la posición final esperada (figura 4.8 posición 3), sigue navegando.
- D. **Cuando detecta un obstáculo, su evasión es lenta.** La evasión del robot es diferente de la prueba anterior, ya que la velocidad de movimiento es menor y la cantidad de rutas posibles también. Cuando detecta un obstáculo (pared en la posición 1, mantel en la 2) se mueve hacia un lado hasta que el obstáculo no aparezca en la imagen.
- E. **La navegación en base al color puede fallar.** Al aplicar las tres restricciones del algoritmo a la posición inicial, el robot hace una navegación simple por el ambiente de prueba. En la posición 3 el robot sigue de frente hacia un obstáculo. Al detener el algoritmo y revisarlo, se observa que el valor de referencia cambia haciendo que el nivel de gris en la columna sea considerado

como referencia de piso haciendo que el robot siga de frente hacia esta (figura 4.9).

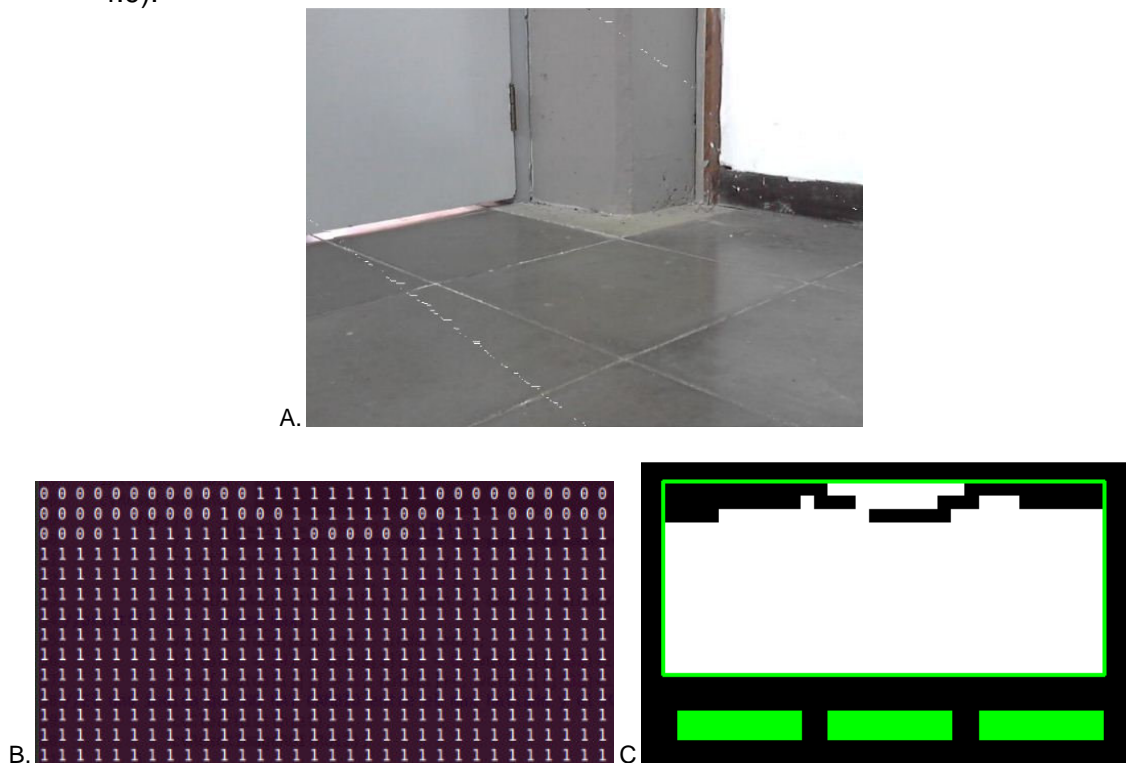


Figura 4.9.A) Imagen captada por la cámara en la posición final del robot. B) Matriz de votación. La cantidad de '1' seguidos permite intuir una gran continuidad de piso. C) Definición de rutas del algoritmo usado, como las tres zonas están en verde (debido a la matriz de votación), el algoritmo elige por defecto seguir de frente. [Fuente propia]

#### 4.4.3. Segundo piso Escuela de Ingeniería Eléctrica y Electrónica. Resultados y análisis.

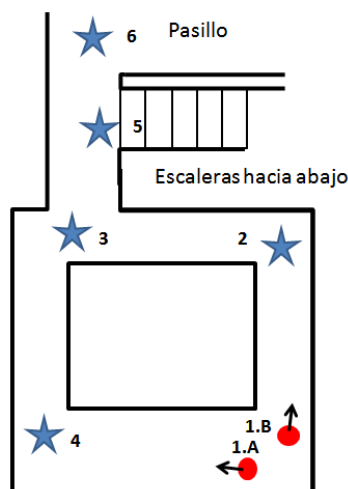
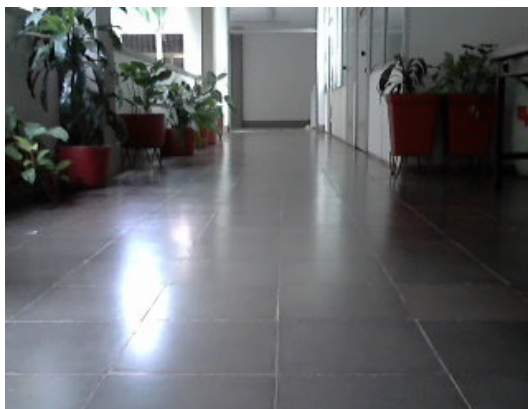


Figura 4.10. El círculo rojo indica la posición inicial del robot en el ambiente y la flecha su orientación. Las estrellas son posiciones donde ocurren cambios en la navegación.

Aspecto	Porcentaje de acierto	Observaciones
Detección de obstáculos	100%, 0%	Detecta obstáculos (100%) con base en la ubicación de estos en la imagen captada por la cámara (literal F), aunque a veces falla (0%, literal G).
Evasión de obstáculos	100%, 50%, 0%	Siempre que el obstáculo se pueda distinguir por su diferente color respecto al piso, la evasión es exitosa (100%). En otros casos, depende de factores externos al robot (50%, 0%), esto se amplía en el literal G, H, I.
Navegación del robot	---	La ausencia de control en la iluminación introduce falsos positivos (literal H), lo que quita confiabilidad al algoritmo.
Evitar riesgo para el robot	100%, 0%	Aunque puede detectar y evadir obstáculos (100%) como se indica en el literal F, I, a veces tiene errores (0%) que llevan al robot a un choque directo referido en literal G, H.

Tabla 4.4. Resultados escenario 2.

- F. **Cuando detecta un obstáculo, su evasión es lenta.** La evasión del robot es diferente de la prueba con el láser, ya que la velocidad de movimiento es menor y la cantidad de rutas posibles también. Cuando detecta un obstáculo se mueve hacia un lado hasta que el obstáculo no aparezca en la imagen. Para el caso de las personas, éstas evaden al robot o se quedan quietas permitiendo al algoritmo detectarlas en la matriz de votación y evadirlas.
- G. **La navegación en base al color puede fallar.** Al aplicar las tres restricciones del algoritmo a la posición inicial (posición 1.A y 1.B), el robot hace una navegación simple por el ambiente de prueba entrando y saliendo de los pasillos. Algunas veces el robot se va de frente hacia un obstáculo (pared en las posiciones 2, 3, 4), al detener el algoritmo y revisarlo, se observa que el valor de referencia cambia haciendo que el nivel de gris sea considerado como referencia de piso haciendo que el robot siga de frente.
- H. **La ausencia de control en las fuentes de luz influye de manera negativa en el algoritmo.** Al contrario de la prueba navegación con el laser y como era de esperarse, las condiciones de luz sí influyen en la navegación para este tipo de sensor. El movimiento del robot no era continuo, pues algunas veces gira ante una ruta libre de obstáculo (posición 6). Cuando se presenta ese cambio de dirección en el robot, se detiene el algoritmo para examinar la matriz de votación hallando que el reflejo de la luz solar en el piso se detecta como agujero creando una RUTA NO SEGURA (figura 4.11). Este efecto reduce su influencia en la tarde dado que la posición del edificio disminuye los reflejos de la luz solar en el piso.



A.) Imagen de prueba.



B.) Componente Y de la imagen de prueba.



C.) ROI\_1 componente Y.

D.) Matriz de votación.

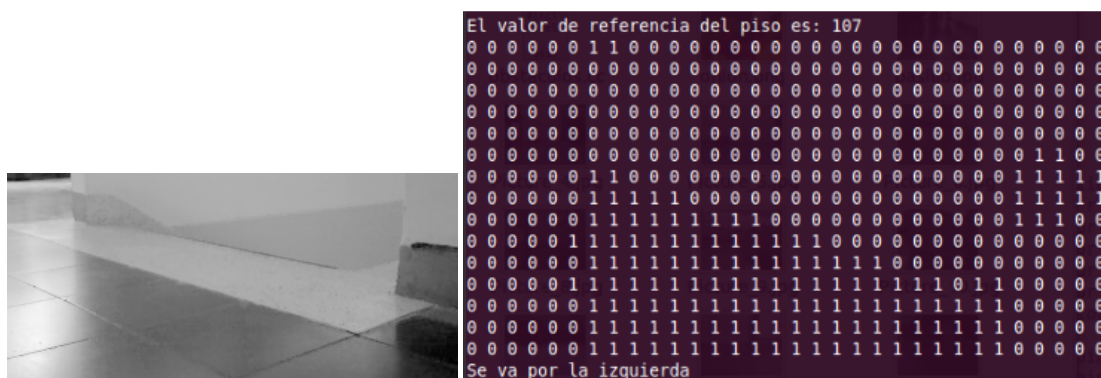
Figura 4.11. C. Influencia de la luz solar en el algoritmo usando la cámara. D.) La matriz de votación contiene '1' que indican un nivel de gris cercano a la referencia (229 para la imagen) y su agrupación describe la continuidad del piso, los '0' indican niveles de gris distintos y se pueden asociar con objetos diferentes al piso; por tanto es un obstáculo. [Fuente propia]

- I. **Puede detectar y evadir la escalera hacia abajo.** A diferencia de la prueba con el láser, los escalones hacia abajo sí son detectados y evadidos (posición 5); sin embargo, su correcta detección puede verse afectada por reflejos de la luz solar en el piso (figura 4.12); así mismo, en algunas ocasiones el giro del robot hace que la cámara no capte la escalera, tomando como ruta los escalones hacia abajo ya que la imagen no contiene las baldosas grises.



A.) Imagen de prueba

B.) Componente Y de la imagen de prueba



C.) ROI\_1 componente Y.

D.) Matriz de votación.

Figura 4.12. Detección y evasión de los escalones hacia abajo. La acumulación de '1' es mayor en la zona izquierda de la imagen comparando con la zona centro y derecha, por lo tanto se elige la mayor concentración de '1' como ruta segura. El algoritmo señala ruta libre a la izquierda. [Fuente propia]



Las pruebas se realizaron cinco veces en cada horario para ambos escenarios. En los dos, se observa el impacto del control en las fuentes de luz. En el segundo escenario (sin control) a pesar de que el robot se desplaza en zig-zag puede detectar los obstáculos con base en el color y evadirlos si el color es diferente al piso (5 detecciones en 5 pruebas, para los tres horarios); pero si presenta colores similares al piso la detección baja (3 de cada 5, para los tres horarios en promedio). El movimiento del robot en el primer escenario (con luz controlada) es mejor que en el segundo, por su desplazamiento continuo y sin cambios abruptos. La detección y evasión de obstáculos con base en el color se comporta similar que en el escenario sin control. Los cambios de iluminación en diferentes horarios influyen fuertemente en el desempeño en el escenario Segundo piso induciendo el algoritmo a errores.

## 4.5. PRUEBA IDENTIFICACIÓN USANDO FUSIÓN SENSORIAL

### 4.5.1. Descripción

Se evalúa la identificación de los obstáculos definidos en la sección 3.2.5 mediante etiquetas, ya que en base a esta etiqueta se hace el cambio en la estrategia de navegación.

La prueba se realiza en el Laboratorio de Robótica móvil para evitar que las personas se atravesasen caminando durante la sesión de prueba. Se coloca al robot (inmóvil) frente a los obstáculos a detectar y se observa la etiqueta de salida. Esta prueba usa el sistema de fusión sensorial que es explicado en el capítulo 3. La iluminación dentro del espacio es controlada.

### 4.5.2. Resultados y análisis.

Verificando el cumplimiento de las condiciones en la sección 3.2.5 para ejecutar la identificación en el bloque fusión sensorial, se colocan al frente del robot los siguientes obstáculos: uno forma plana (figura 4.9), una caneca para residuos sólidos (figura 4.10), una elipse (figura) y las escaleras hacia abajo. También se hacen simulaciones usando el software Stage para evaluar el sistema con otros obstáculos no presentes en el ambiente estructurado.

Obstáculo	Porcentaje de acierto	Observaciones
Pared, tablero	100%, 80%	Los porcentajes varían dependiendo de la ubicación del obstáculo en el FoV de fusión; es decir, dentro del rango 60° - 120° del láser ya que incluye el FoV de la cámara es más alto y cerca de los límites (60° ó 120°) es menor.
Círculo, caneca de residuos sólidos	90%, 70%	
Elipse	90%, 70%	
Escalera	100%, 60%	Para el caso de la escalera, depende de la posición del robot frente a ella y el complemento que le pueda brindar el sensor láser.

Tabla 4.5. Resultados de la prueba.

- A. **La respuesta del sistema es la esperada.** Se coloca cada obstáculo al frente del robot y se ejecuta el algoritmo 10 veces. Como resultado, la etiqueta esperada es igual a la respuesta del sistema (figura 4.13, 4.14 y 4.15). La cantidad de aciertos en identificación es 9 (en promedio) de 10 pruebas

independiente de la posición del obstáculo cuando se encuentra en la parte central del FoV de ambos sensores. Cuando el obstáculo se encuentra más cerca de los límites donde ocurre el cruce del FoV de ambos sensores, los aciertos en la identificación son 7 (en promedio) de 10 pruebas.

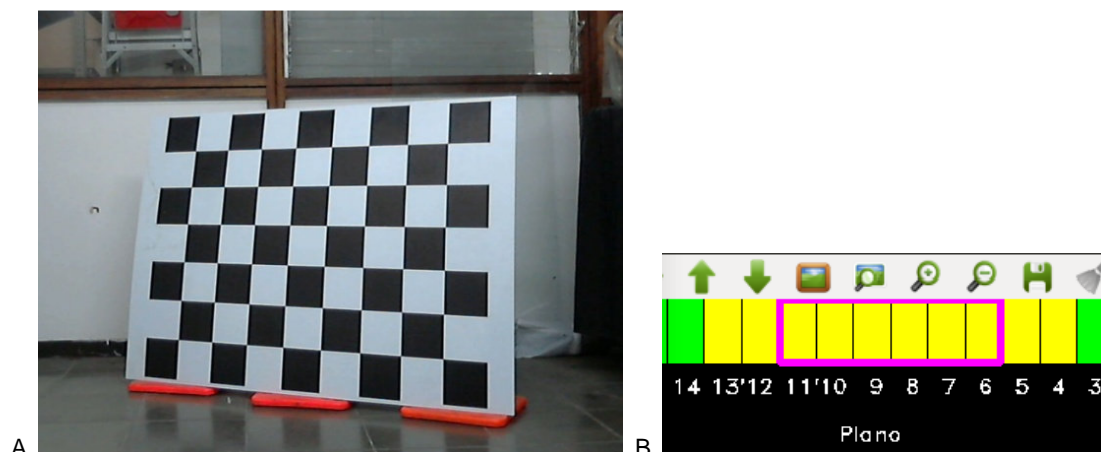


Figura 4.13. A.) Obstáculo en el escenario real. La imagen B.) corresponde a la etiqueta de identificación del sistema de fusión. [Fuente propia]

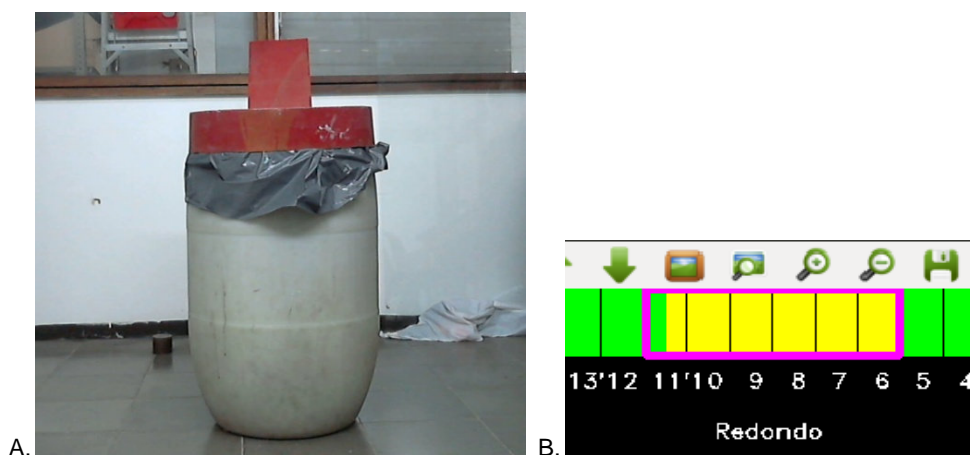


Figura 4.14. A.) es un obstáculo en el escenario real. B.) muestra la salida del sistema de fusión cuya etiqueta identifica al obstáculo correctamente. [Fuente propia]

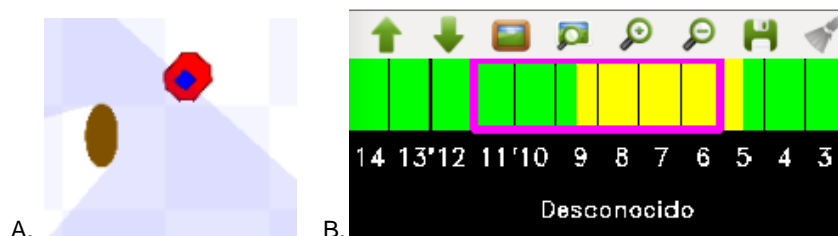


Figura 4.15. A.) muestra la simulación de un objeto con forma de elipse y B.) es indica la etiqueta de salida correspondiente. [Fuente propia]

B. **El sistema permite al robot evadir la escalera.** En el momento que el robot se encuentra de frente a los escalones hacia abajo (figura 4.16), el sistema responde indicando la etiqueta de escalera. No obstante, la identificación se



dificulta cuando el robot se acerca por el lado de la pared como si indica en la figura 4.16 mediante estrellas. Sin embargo, la cercanía con la pared hace que el robot la detecte (debido al riesgo 5) y desvíe su camino alejándose de los escalones hacia abajo.

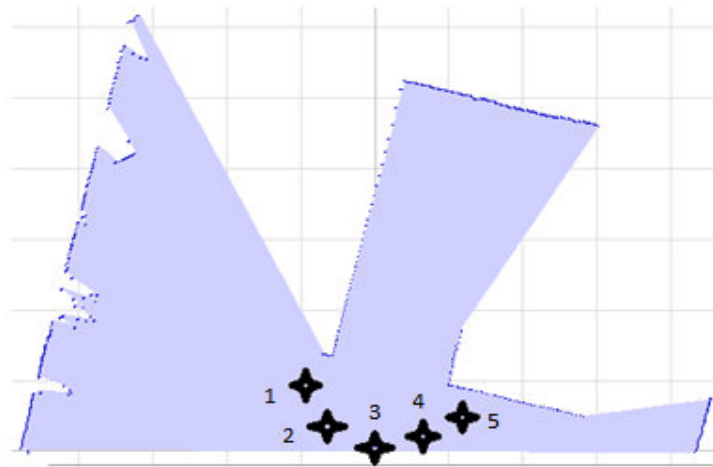


Figura 4.16. Las estrellas indican la posición de robot frente a las escaleras. [Fuente propia]

El sistema de fusión tiene un porcentaje de acierto en la identificación de obstáculos entre el 70% y 90% (en promedio) siempre y cuando se encuentren en el FoV de ambos sensores; por fuera del FoV de la cámara y dentro del FoV del láser, sólo se detectan mediante riesgos y su evasión es tomar la dirección contraria a la ubicación del obstáculo.

En la prueba para la escalera, la posición del robot en 2, 3 y 4 (figura 4.16) presenta una identificación positiva total: 10 etiquetas correctas en 10 pruebas. En contraste, para las posiciones 1 y 5 el total de etiquetas correctas baja entre 6 y 7 debido a que la imagen capta pocos escalones hacia abajo lo cual disminuye la cantidad de líneas Hough detectadas. Sin embargo, la cercanía con la pared aumenta el riesgo a 5 tomando la acción evasiva básica, evitando la escalera sin necesidad de detectarla. Lo anterior es una muestra de la complementariedad entre ambos sensores.

## 4.6. PRUEBA NAVEGACIÓN MEDIANTE FUSIÓN SENSORIAL

### 4.6.1. Descripción

Esta prueba evalúa la definición de trayectoria del sistema de fusión sensorial mediante la navegación del robot en el ambiente de prueba.

Se hace una simulación y una prueba real con la misma ubicación para los obstáculos. El escenario es el segundo piso de la Escuela de Ingeniería Eléctrica y Electrónica.

Con base a las pruebas de navegación con el láser y navegación con cámara, se elige como horario de prueba la tarde por tener la mejor respuesta para la cámara puesto que para el sensor láser, es indiferente el horario. La prueba se realiza 10 veces cambiando los obstáculos de ubicación.

#### 4.6.2. Resultados y análisis.

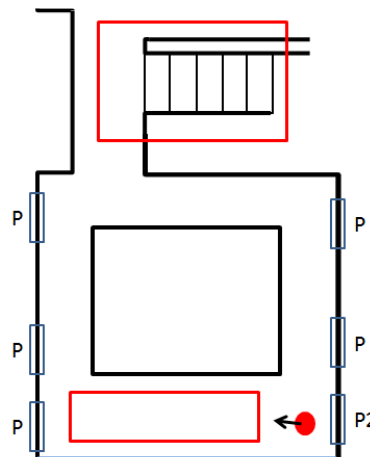


Figura 4.17. Escenario de la prueba. El recuadro rojo indica la zona donde se ubican los obstáculos en el escenario. [Fuente propia]

Aspecto	Porcentaje de acierto	Observaciones
Detección de obstáculos	95%, 75% (en promedio)	La detección de obstáculos tiene base en la prueba de identificación, de allí los porcentajes. Los obstáculos son detectados e identificados por el sistema de fusión. Los distintos obstáculos se pueden detectar, a excepción de la tabla de la mesa, como se explica en el literal D. Para los demás, se relacionan los literales A, B, C.
Evasión de obstáculos	100%	El sistema cuenta con cuatro categorías de clasificación que agrupan los obstáculos presentes en el ambiente de prueba (literal B). Cada obstáculo se evade de manera diferente como lo indica el literal C.
Navegación del robot	---	Mitiga la sensibilidad de la cámara ante el control de luz (literal A) y la debilidad del láser en la detección de la escalera. Dependiendo de la ubicación del obstáculo, su distancia e identificación, se elige la estrategia de navegación adecuada, como se amplía en el literal C.
Evitar riesgo para el robot	100%	Se puede detectar, identificar y evadir obstáculos con alto riesgo como la escalera y el vidrio, aparte de los demás.

Tabla 4.6. Resultados de la prueba.

- A. **La ausencia o presencia de control en las fuentes de luz no influyen en el sistema de fusión.** A diferencia de la prueba navegación con cámara donde la luz causa errores, en la extracción de características no influye de manera significativa ya que los reflejos de luz en el piso no aporta líneas que afecten el cálculo de la transformada Hough. Además, el sistema puede detectar la escalera hacia abajo y evadirla puesto que el bloque de identificación de obstáculos la etiqueta como tal.
- B. **Los obstáculos presentes en el escenario se pueden asociar con alguna de las cuatro categorías.** La extracción de características depende de lo que realicen los bloques asignados a los sensores. Así se obtienen los patrones de entrada para la RNA que los procesa y luego esta entrega el resultado como una etiqueta que identifica el obstáculo detectado. Se tienen cuatro categorías que engloban la mayoría de casos observados en el ambiente de prueba.

Aunque en algunos casos la etiqueta no corresponde con el obstáculo detectado (20%), este es evadido.

- C. **Los obstáculos no se evaden de la misma manera.** La ubicación y distancia del obstáculo respecto al robot no es el único criterio que se usa para evadir aquellos que se detectan, también se usa su forma geométrica para definir una estrategia de navegación adecuada como se describe en la tabla 3.4 del capítulo 3, ya que se puede hacer algo distinto a solamente alejar el robot por el lado contrario.
- D. **Dificultad en la evasión de las mesas.** Respecto a las mesas altas, no hay diferencia entre el sistema de fusión y el sensor láser solo, ya que la mesa está por encima del haz láser y la posición de la cámara no permite capturar la mesa totalmente en la imagen (apenas aparecen las patas). Para sortear esta dificultad, se hace lo mismo que en la prueba navegación con el láser: colocar un cono plástico para indicar que ese espacio bajo la mesa no se puede transitar.

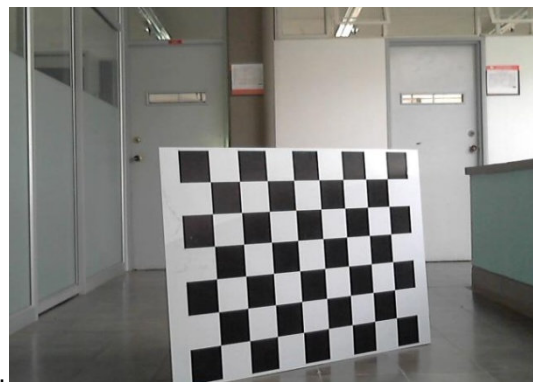
Una forma de evaluar el desempeño del sistema de fusión es desde el resultado. Para el caso del vidrio y la escalera hacia abajo es satisfactoria porque los detecta, identifica y evade. Es de anotar que en el escenario de prueba, el vidrio presente está cubierto por una envoltura de color opaco y permite su detección como obstáculo plano (lado izquierdo figura 4.18). Para los demás obstáculos, la etiqueta permite su identificación y evasión.



A.



B.



C.

Figura 4.18. A.) imagen del ambiente de prueba sin obstáculos. B.) C.) Muestran el mismo ambiente con un obstáculo. [Fuente propia]



Figura 4.15. Imágenes del robot navegando entre pasillos. [Fuente propia]

#### 4.7. ANÁLISIS GENERAL DEL SISTEMA.

Un sistema de fusión sensorial, en general, debe evaluar su desempeño como conjunto respecto a cada sensor que se utiliza. De esa manera, se aplica la ecuación 4 del capítulo 2.

A partir de las pruebas navegación con el láser y navegación con cámara, se obtiene el desempeño  $p_m$  para cada sensor. Con las pruebas de identificación y navegación mediante fusión sensorial, se consigue el desempeño  $p_f$ .

$$p_F \ll \min_m(p_m) \quad \text{Ecuación 4.}$$

Si el sistema tiene un desempeño menor comparado con el mínimo desempeño de alguno de los sensores, la fusión desarrollada no aporta ninguna mejoría y por lo tanto, se debe descartar y cambiarla.

Pero si el desempeño de la fusión sensorial supera al máximo valor de cualquiera de los sensores utilizados, la fusión desarrollada logra mejorar la representación de la información obtenida del ambiente, reducir la incertidumbre en la medida realizada por el sensor, reducir el tiempo de procesamiento de la información o mitigar los efectos de los disturbios que ingresen al sistema.

En este trabajo de grado: se logró una mejoría en la representación de la información obtenida del ambiente ya que se puede identificar el obstáculo que es detectado y elegir una adecuada estrategia de evasión, también se logró mitigar los efectos de la ausencia de control en las fuentes de luz que perturban al sensor cámara.

En suma, la mejor prueba para valorar el sistema de fusión sensorial desarrollado, es la respuesta del robot mientras hace navegación **wandering** en el escenario de la prueba navegación mediante fusión sensorial.

A continuación, se realiza un cuadro comparativo del sistema de fusión sensorial desarrollado con los algoritmos de navegación, seguido de los alcances y las limitaciones.

Aspecto	[Borenstein, Koren 91] Vector field histogram – fast obstacle avoidance for mobile robots	[Zhan, Huang, Wu 08] Automatic navigation for a mobile robot with monocular vision	Identificación y evasión de obstáculos mediante fusión sensorial para robots móviles.
<b>Detección de obstáculos (% acierto)</b>	100% Plano, Redondo. 0% Escalera.	100% Color obstáculo distinto del piso. 0% Color obstáculo igual.	95%, 75% En promedio las cuatro categorías
<b>Criterios utilizados en la detección</b>	Ubicación y distancia de obstáculos respecto al robot.	Ubicación del obstáculo.	Ubicación y distancia del obstáculo respecto al robot. Forma geométrica del obstáculo.
<b>Evasión de obstáculos (% de evasión)</b>	100% Cuando es detectado 0% Cuando no lo es.	100% Si es detectado. 50% Depende de la prueba 0% No lo es.	100%
<b>Estrategia de evasión</b>	Igual para todos los obstáculos: alejarse de ellos.	Ubicarlo en la imagen y elegir la ruta contraria hasta que desaparezca de la imagen.	Depende de la identificación y la distancia, así elige la estrategia más adecuada.
<b>Navegación del robot</b>	Fluida y continua.	Depende del control de las fuentes de luz.	Fluida y continua.
<b>Seguridad para el robot</b>	En algunos puntos del ambiente de prueba no protege al robot.	Depende del control de las fuentes de luz y de la diferencia de color entre los obstáculos y el piso.	Falla solamente cuando se encuentra con una mesa alta como obstáculo.

Tabla 4.7. Cuadro comparativo.

**Alcance:** El sistema de fusión permite al robot navegar en el ambiente de prueba mientras detecta, identifica y evade obstáculos como: paredes, puertas (etiqueta plano), canecas para residuos sólidos, materas (etiqueta redondo), patas de mesa, personas (riesgo 5), y los de mayor riesgo como los escalones hacia abajo (etiqueta escalera) y el vidrio (por riesgo o etiqueta).

Comparando con los algoritmos de navegación que utilizan los sensores por separado, el sistema de fusión sensorial desarrollado además de detectar el obstáculo, puede identificarlo mediante una etiqueta y con base en ello, determinar cuál estrategia de navegación es la más adecuada para evadirlo.

La ausencia de control en las fuentes de luz que afecta a la navegación usando cámaras, es mitigada por el sistema de fusión. Además, la complementariedad que se usa en el sistema entre el sensor láser y la cámara, evita que el robot detecte reflejos de luz en piso como obstáculos y columnas (también paredes) como una ruta libre.

El obstáculo con mayor amenaza para el robot, la escalera hacia abajo, es detectado, identificado y evadido garantizando la navegación segura del robot en el escenario de prueba Segundo piso Escuela de Ingeniería Eléctrica y Electrónica.

**Limitaciones:** Respecto a las mesas altas, aunque se pueden detectar sus patas no es posible detectar la tabla que puede detener el robot (por la altura de la estructura montada sobre él) dado que están por fuera de los **FoV** de la cámara y del láser.

Debido a que el algoritmo no cuenta con una “memoria del recorrido”, puede pasar varias veces por el mismo lugar.

Finalmente, se muestra una tabla resumen del sistema de fusión desarrollado.

Características	Rango	Condiciones de operación
Detección de obstáculos por categorías identificación de estos por etiquetas y elección de la más adecuada estrategia de navegación para evadirlo.	El cruce entre el FoV de la cámara y el sensor láser crea tres regiones: 0° - 60° y 120° - 180° solamente con el sensor láser. En la región 60° - 120° es donde se cruzan los dos sensores y se realiza la extracción de características.	Aunque el horario no es determinante, el mejor desempeño es 2 pm – 5 pm, ya que el control o no en la iluminación no afecta al sistema y además, la afluencia de personas que transita por el ambiente de prueba es mínima.
La navegación a través de los pasillos es fluida y sin cambios abruptos de dirección.		
La navegación que hace es <b>wandering</b> .		Cumplimiento de las tres restricciones del capítulo 3, sección 3.2.4.

Tabla 4.8. Condiciones de operación.

#### 4.8. CONCLUSIONES

- En las pruebas navegación con el láser y navegación con cámara se evaluaron **los desempeños *pm* de cada sensor**, lo que aportó información para la aplicación de la ecuación que mide la efectividad de la fusión sensorial y también resaltó las diferencias y limitaciones que cada sensor posee debido a su naturaleza: uno mide distancia y el otro tanto la intensidad de luz como el color.
- En las pruebas de identificación y navegación mediante fusión sensorial se obtuvo **el desempeño *pf* del sistema de fusión**, lo que contribuyó con información para la aplicación de la ecuación que mide la efectividad de la fusión sensorial revelando una reducción del 70% del error en la detección y del 90% en la evasión de obstáculos, también la complementariedad de los distintos sensores (uno cubre las debilidades del otro) hizo que el sistema muestre un mejor aspecto en **representación de la información y robustez**.
- La identificación de obstáculos con mayor riesgo como los escalones hacia abajo y el vidrio (mediante etiquetas), y la mitigación de los efectos debido a la ausencia de control en las fuentes de luz en el ambiente de prueba, son una muestra de la **robustez del sistema**.
- Los diferentes valores en la detección acertada de obstáculos se deben a casos que ocurren en cada prueba exponiendo una limitación del algoritmo evaluado allí, ya que la detección cambia de acuerdo con la posición del obstáculo frente al robot, si aparece dentro del FoV de los sensores o se mueve rápido.
- Cuando el porcentaje de acierto en la detección varía, el porcentaje de acierto en la evasión también. Lo que muestra una relación directa entre la detección y la evasión.
- El sistema de fusión desarrollado protege de manera implícita la integridad del robot al ser capaz de evadir obstáculos que los algoritmos de navegación usando cada sensor por separado no pueden detectar.

## 5. CONCLUSIONES

- Se logró el desarrollo y la implementación de un sistema de fusión sensorial entre un sensor láser y una cámara monocular utilizando la técnica de Redes Neuronales Artificiales. El sistema implementado permitió la detección e identificación de obstáculos utilizando cuatro categorías en un ambiente interior tipo oficina, y cambiar la estrategia de navegación con base en el tipo de obstáculo.
- El sistema de fusión sensorial desarrollado mejoró el porcentaje de acierto en la detección al ubicarlo en el rango de 100% - 70% comparado con la mejor detección usando los sensores por separado. Asimismo, el porcentaje de acierto en la evasión es de 100% con base en el porcentaje de detección del sistema.
- Se logró identificar obstáculos y generar comandos de movimiento del robot con una sola RNA con cuatro características suministradas por la cámara y por el láser.
- La adición del criterio de la forma geométrica a los criterios de la distancia y ubicación de obstáculos mejoró el desempeño del sistema de fusión desarrollado para la navegación del robot.
- Se logró desarrollar un sistema de fusión sensorial que mejora **la representación de información**, además mostró una **robustez de sistema** ya que se pueden detectar, identificar y evadir no solo obstáculos estáticos y dinámicos, sino aquellos que representan mayor peligro para el robot como los escalones hacia abajo y el vidrio.

## 6. PERSPECTIVAS FUTURAS DE TRABAJO

El propósito de los sensores externos es proveer al sistema de información útil respecto a algunas características de interés en el ambiente. En paralelo, la potencial ventaja en la fusión sensorial es obtener información más precisa, hallar otra característica que los sensores por sí solos no entregan, o disminuir sea el costo o el tiempo del procesamiento. Además, el uso de modelos del mundo permite al sistema de fusión almacenar y ‘razonar’ la información que le llega **[Luo, Kay 89]**.

Por lo anterior, una perspectiva de trabajo futuro es añadir un bloque de creación de mapa y evaluación de posición. Con esto, también se puede señalar en el mapa los lugares recorridos, desconocidos y aquellos con riesgo alto, medio o bajo para la navegación.

Los tres aspectos clave para la navegación: detectar y evadir los obstáculos en el ambiente, medir la distancia entre los obstáculos y el robot, y elegir una ruta segura **[Zhan, Huang, Wu 08]** se pueden mejorar si es añadido un bloque de predicción de movimiento en los obstáculos dinámicos.

Muchos de los métodos de fusión sensorial hacen suposiciones explícitas o implícitas sobre la información que entregan los sensores **[Luo, Kay 89]**. Para buscar complementar la información de la clasificación de los obstáculos se puede cambiar la posición de la cámara cruzando su **FoV** con el del láser logrando un aporte a las características obtenidas por el láser, utilizando la información del color.

Los algoritmos de navegación utilizados para las comparaciones son algoritmos que se tomaron como base para la creación de otros. Así, del algoritmo VFH aparece VFH+, luego VFH ★, y el algoritmo VFH se presenta como demo y opción de navegación autónoma en el software Player. Este algoritmo se toma como canon para realizar comparaciones con el sistema de fusión sensorial. El algoritmo **[Zhan, Huang, Wu 08]** aplica el criterio más ampliamente utilizado en la navegación con cámaras: ubicación del obstáculo. Se propone como trabajo futuro comparar con otros algoritmos que no sean canon (mallas de ocupación) y que usen otros criterios para navegar utilizando cámaras (punto de fuga, distancia por estereovisión, segmentación del camino).



## 7. REFERENCIAS BIBLIOGRÁFICAS

[Akihisa, Kosaka, Kak 98]

Akihisa, Ohya. Kosaka, Akio. Kak, Avinash. Vision-based navigation by a mobile robot with obstacle avoidance using single camera vision and ultrasonic sensing. IEEE Transactions on Robotics and Automation. 1998.

[Baltzakis, Argyros, Trahanias 03]

Baltzakis, Haris. Argyros, Antonis. Trahanias, Panos. Fusion of laser and visual data for robot motion planning and collision avoidance. Springer-Verlag. 2003

[Bardyn 84]

J. J. Bardyn. Une architecture VLSI pour un opérateur de filtrage median. Congrès reconnaissance des formes et intelligence artificielle (vol. 1, pp. 557–566). Paris. 25–27 January 1984

[Borenstein, Koren 91]

Borenstein, J. Koren, Y. The Vector Field Histogram – Fast obstacle avoidance for mobile robots. IEEE Journal of Robotics and Automation Vol 7, No 3. 1991.

[Bradski, Kaehler 08]

Gary Bradski, Adrian Kaehler. Learning OpenCV. O'Reilly Media, Inc. Primera Edición. 2008

[Canny 86]

J. Canny. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1986): 679–714

[Castillo, Sáez 03]

M. A. Castillo, J. M. Sáez. Modelo de sonar de largo alcance basado en tecnología estéreo. Universidad de Alicante. 2003.

[Duda, Hart 72]

R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. Communications of the Association for Computing Machinery 15 (1972): 11–15.

[Flynn 88]

Combining sonar and infrared sensors for mobile navigation. International Journal of Robotics Research. Páginas 5 – 14, Diciembre 1988.

[Gálvez, Sjö, Chandana, Jensfelt 08]

Gálvez López, Dorian. Sjö, Kristoffer. Chandana, Paul. Jensfelt, Patric. *Hybrid Laser and Vision based object search and localization*. Centre for autonomous Systems, Royal Institute of Technology. IEEE International conference on Robotics and Automation. 2008.

[Jaisakthi, Chandrabose 11]

Jaisakthi, S. M. ChandraboseAravindan. *Face Detection using Data and Sensor fusion techniques*. Department of computer science and engineering SSN College of engineering. Chennai, India. International conference of soft computing and pattern recognition.

[Laganière 11]

Robert Laganière. *OpenCV 2 Computer Vision Application Programming Cookbook*. Packt Publishing Ltda. 2011

[Liggins, Hall, Llinas 09]

Martin E. Liggins, David L. Hall, James Llinas. *Handbook of Multisensor data fusion, theory and practice*. Editorial CRC Press. 2da Edición. 2009

[Luo, Kay 89]

Luo, Ren C. Kay, Michael G. *Multisensor integration and fusion in intelligent systems*. IEEE transactions on system, man and cybernetics. Vol 19 No 5. Septiembre – Octubre 1989.

[Mitchell 10]

H. B. Mitchell. *Fusion: Concepts and ideas*. Editorial Springer. 2010.

[MobileRobots 06]

Pioneer 3 Operations Manual with MobileRobots Exclusive, Advanced Robot Control & Operation Software. 2006. Mobile Robots Inc.

[Moshe, Zhu, Kalata 97]

Moshe Kam, Xiaoxun Zhu, Paul Kalata. *Sensor fusion for mobile robot navigation*. Proceedings of the EIEE, Vol 85 No. 01, Enero 1997.

[Mühlbauer, Kühnlenz, Buss 08]

Mühlbauer, Quirin. Kühnlenz, Kolja. Buss, Martin. *Fusing Laser and Vision data with a genetic ICP algorithm*. Technische Universität München. 2008.

### [\[@OpenCVDocumentation\]](#)

<http://docs.opencv.org/2.4/index.html> Aunque pueden salir varias versiones como la 2.4.0, 2.4.1, 2.4.2 hasta la 2.4.6, la diferencia entre éstas se encuentra en la forma de programar los métodos sin afectar su formato para ser invocados por el usuario, o bien se agrega otra forma de invocarlos, o se agrega otro lenguaje de programación en el cual se puedan usar. Open CV originalmente fue diseñado en C/C++, ahora ya está en versiones instalables en Java y Python haciendo compatible casi que en cualquier plataforma: Linux, Windows, android.

[Owen 10]

Jennifer Owen. How to use Player/Stage.2Da Edición. 16 de Abril de 2010.

### [\[@Player\]](#)

<http://playerstage.sourceforge.net/doc/Player-3.0.2/player/index.html> Contiene la página principal del proyecto en su versión 3.0.2, desde la cual se puede acceder a los proxy dando click en la secuencia: *Clientlibraries*, *libplayerc++*, *Proxies*. Los comandos que usa Player en el enlace *Utilities*. Los driver con *Supported devices* o también por *libplayerdrivers*, *drivers*.

### [\[@ Proxies\]](#)

[http://playerstage.sourceforge.net/doc/Player-3.0.2/player/group\\_player\\_clientlib\\_cplusplus\\_proxies.html](http://playerstage.sourceforge.net/doc/Player-3.0.2/player/group_player_clientlib_cplusplus_proxies.html) Es la ruta directa hacia los Proxies que se pueden controlar con Player.

[Roberto, Machado, Pinto, Branco 13]

Roberto, Guilherme Freire. Pinto, A. R. Machado, J. M. Universidade Estadual Paulista. Branco, KalinkaCastelo. Universidade de Sao Paulo. Local data fusion algorithm for fire detection through mobile robot. 2013.

[SICK AG Division]

Laser Measurement System LMS 200/ LMS 211/ LMS 220/ LMS 221/ LMS 291.SICK AG Division Auto Ident. Germany.

[Shuihua, Tingli 12]

Shuihua, Wang. Yingli, Tian. Detecting stairs and pedestrian crosswalk for the blind by RGBD camera.IEEE International ConferenceonBioinformatics and BiomedicineWorkshops. 2012.

### [\[@Stage\]](#)

<http://playerstage.sourceforge.net/doc/Stage-3.2.1> Es la pagina principal de la versión 3.2.1 del Stage, muy similar a la versión 3.2.2 que se usa en este trabajo de grado, de la cual se puede acceder a toda la información necesaria para manejar mas hábilmente el simulador Stage. Los enlaces *World*, *WorldGUI* me permiten asignar valores a los parámetros que controlan la creación del mundo que genera la

simulación. *Model* contiene unas guías sobre cómo generar y configurar los modelos y objetos que aparecerán en el mundo creado.

[Vaughan, Gerkey 06]

Richard T. Vaughan, Brian P. Gerkey. Really Reusable Robot Code and the Player/Stage Project. Software engineering of experimental robotics, Springer Tracts on Advanced Robotics, D. Brugali Editor, Springer, 2006

[Wei, Lu 06]

Wei Jiang, Lu Jian. *Panoramic 3D reconstruction by fusing color intensity and laser range data*. International Conference on Robotics and Biomimetics. 2006

[Whitbrook 10]

Amanda Whitbrook. Programming mobile robots with Aria and Player. Editorial Springer. 2010.

[@ Willow garaje]

<http://www.willowgarage.com> Contiene la información de los proyectos que soporta y las actualizaciones y entregas de los software que diseña (OpenCV, ROS, Player, Stage, Gazebo, librerías), también del hardware referente a robótica, como el robot PR2.

[Woods, Eddins 09]

Gonzalez, Rafael. Woods, Richard. Eddins, Steven. Digital Image Processing using Matlab. Gatesmark Publishing. 2009.

[Xavier, Pacheco, Castro, Ruano, Nunes 04]

Xavier, Joao. Pacheco, Marco. Castro, Daniel. Ruano, António. Nunes, Urbano. Fast Line, Arc/Circle and Leg detection from Laser scan data in a Player Driver. Institute of System and Robotics University of Coimbra, Centre for Intelligent System University of Algarve, Portugal. 2004.

[Zhan, Huang, Wu 08]

Zhan, Qiang. Huang, Shouren. Wu, Jia. Automatic Navigation for a Mobile robot with monocular vision. Robotics and Automation Magazine 2008.

## 8. ANEXOS

En esta sección, se consignan los códigos de ejemplo que son guía para la configuración del robot, solicitud y recepción de datos a los sensores usando Player, crear una Machine Learning con OpenCV y un ejemplo de calibración para la cámara con OpenCV.

Archivo de configuración del pioneer para la versión Player 3.0.2. Tomado de **[Whitbrook 10]**

```
driver(
  name "p2os"
  provides ["odometry::position2d:0"
    "sonar:0"
    "power:0"
    "bumper:0"
    "gripper:0"]
  port "/dev/ttyS0"
)

driver (
  name "sicklms200"
  provides ["laser:0"]
  port "/dev/ttyS2"
  rate 38400    #Relativo al baud rate
  delay 32      #En seg, despues del encendido
  resolution 50 #50 es 361 lecturas y 100 es 181
  range_res 10  #En mm
  alwayson 1
)

driver(
  name "camerauvc"
  provides ["camera:0"]
  port "/dev/video0"
  size [640 480]
)
```

Código de ejemplo para la conversión de datos entre la cámara y el Servidor Player, para facilitar el procesamiento de la imagen usando OpenCV. Aquí se muestra la solicitud y recepción de datos para el sensor cámara.

```
#include </usr/local/include/opencv2/opencv.hpp>
#include <iostream>
#include <stdio.h>
#include </usr/local/include/player-3.0/libplayerc++/playerc++.h>
#include <time.h>
#include <stdlib.h>
#include <unistd.h>
```

```

using namespace PlayerCc;

int main(int argc, char* argv[]){
    using namespace PlayerCc;
    PlayerClient robot1("192.168.2.100", 6665);
    Position2dProxy p2dp(&robot1, 0);
    CameraProxy cp(&robot1, 0);
    int hig, wid, dep;
    p2dp.SetMotorEnable(true);
    p2dp.ResetOdometry();
    robot1.Read();
    dep = cp.GetDepth();
    wid = cp.GetWidth();
    hig = cp.GetHeight();
    uint8_t imgBuffer[dep*wid*hig];
    IpLImage* in = cvCreateImage(cvSize(wid, hig), dep/3, 3);

    while(true){

        robot1.Read();
        cp.Decompress();
        cp.GetImage(imgBuffer);
        for (int i=0; i<wid; i++) {
            for(int j=0; j<hig; j++){
                in->imageData[wid*j*3 + i*3 + 0] = (char)imgBuffer[wid*j*3 + i*3 + 2];
                in->imageData[wid*j*3 + i*3 + 1] = (char)imgBuffer[wid*j*3 + i*3 + 1];
                in->imageData[wid*j*3 + i*3 + 2] = (char)imgBuffer[wid*j*3 + i*3 + 0];
            }
        }

        IpLImage* img = cvCreateImage(cvSize((int) in->width, (int) in->height, in->depth, 3);
        IpLImage* rojo = cvCreateImage(cvSize((int) in->width, (int) in->height, 8, 1);
        IpLImage* azul = cvCreateImage(cvSize((int) in->width, (int) in->height, 8, 1);
        IpLImage* verde = cvCreateImage(cvSize((int) in->width, (int) in->height, 8, 1);

        cvCvtColor(in, img, CV_RGB2HSV);
        cvSplit(img, rojo, azul, verde, 0);
        cvShowImage("R_G_B", in);
        cvShowImage("Hue", rojo);
        cvShowImage("Saturacion", azul);
        cvShowImage("Intensidad", verde);

        char c = cvWaitKey(25);
        if (c == 27) break;
        cvReleaseImage(&img); cvReleaseImage(&rojo); cvReleaseImage(&verde);
        cvReleaseImage(&azul);
    }
    printf("El ancho de la imagen IN es: %i , y el alto: %i , y la profundidad es %i.\n", in->width,in-
    >height,in->depth);
    cvReleaseImage(&in);
    return 0;
}

```

El siguiente es el código usado en la calibración de la cámara. Muestra algunas funciones usadas por OpenCV. Tomado de **[Bradski, Kaehler 08]**

```

#include <stdio.h>
#include <stdlib.h>
#include </usr/local/include/opencv2/opencv.hpp>

```

```

int n_boards = 0;
const int board_dt = 20;
int board_w;
int board_h;
int main(int argc, char* argv[]) {
    if(argc != 4){
        printf("ERROR: Cantidad equivocada de parametros de entrada\n"
            "1 esquinas internas del tablero a lo ancho\n"
            "2 esquinas internas del tablero a lo alto\n"
            "3 imagenes validas para la calibracion\n"
            "Se recomienda que uno sea par y el otro impar, para distinguir alto de ancho.\n"
        );
        return -1;
    }
    board_w = atoi(argv[1]);
    board_h = atoi(argv[2]);
    n_boards = atoi(argv[3]);
    int board_n = board_w * board_h;
    CvSize board_sz = cvSize( board_w, board_h );
    CvCapture* capture = cvCreateCameraCapture( -1 );
    assert( capture );
    cvNamedWindow( "Calibration" );

    CvMat* image_points = cvCreateMat(n_boards*board_n,2,CV_32FC1);
    CvMat* object_points = cvCreateMat(n_boards*board_n,3,CV_32FC1);
    CvMat* point_counts = cvCreateMat(n_boards,1,CV_32SC1);
    CvMat* intrinsic_matrix = cvCreateMat(3,3,CV_32FC1);
    CvMat* distortion_coeffs = cvCreateMat(5,1,CV_32FC1);
    CvMat* rot_vec = cvCreateMat(3,1,CV_32FC1);
    CvMat* trans_vec = cvCreateMat(3,1,CV_32FC1);
    CvPoint2D32f* corners = new CvPoint2D32f[ board_n ];
    int corner_count;
    int successes = 0;
    int step, frame = 0;

    IplImage *image = cvQueryFrame( capture );
    IplImage *gray_image = cvCreateImage(cvGetSize(image),8,1);

    while(successes < n_boards) {
        if(frame++ % board_dt == 0) {
            int found = cvFindChessboardCorners( image, board_sz, corners, &corner_count,
                CV_CALIB_CB_ADAPTIVE_THRESH | CV_CALIB_CB_FILTER_QUADS);
            cvCvtColor(image, gray_image, CV_BGR2GRAY);
            cvFindCornerSubPix(gray_image, corners, corner_count, cvSize(11,11),cvSize(-1,-1),
                cvTermCriteria(CV_TERMCRIT_EPS+CV_TERMCRIT_ITER, 30, 0.1 ));
            cvDrawChessboardCorners(image, board_sz, corners,corner_count, found);
            cvShowImage( "Calibration", image );
            if( corner_count == board_n ) {
                step = successes*board_n;
                for( int i=step, j=0; j<board_n; ++i,++j ) {
                    CV_MAT_ELEM(*image_points, float,i,0) = corners[j].x;
                    CV_MAT_ELEM(*image_points, float,i,1) = corners[j].y;
                    CV_MAT_ELEM(*object_points,float,i,0) = j/board_w;
                    CV_MAT_ELEM(*object_points,float,i,1) = j%board_w;
                    CV_MAT_ELEM(*object_points,float,i,2) = 0.0f;
                }
                CV_MAT_ELEM(*point_counts, int,successes,0) = board_n;
                successes++;
            }
        }
    }
}

```

```

int c = cvWaitKey(15);
if(c == 'p'){
    c = 0;
    while(c != 'p' && c != 27){
        c = cvWaitKey(250);
    }
}
if(c == 27)
return 0;
image = cvQueryFrame( capture );
} //End del while

CvMat* object_points2 = cvCreateMat(successes*board_n,3,CV_32FC1);
CvMat* image_points2 = cvCreateMat(successes*board_n,2,CV_32FC1);
CvMat* point_counts2 = cvCreateMat(successes,1,CV_32SC1);

for(int i = 0; i<successes*board_n; ++i) {
    CV_MAT_ELEM( *image_points2, float, i, 0) = CV_MAT_ELEM( *image_points, float, i, 0);
    CV_MAT_ELEM( *image_points2, float,i,1) = CV_MAT_ELEM( *image_points, float, i, 1);
    CV_MAT_ELEM(*object_points2, float, i, 0) = CV_MAT_ELEM( *object_points, float, i, 0) ;
    CV_MAT_ELEM( *object_points2, float, i, 1) = CV_MAT_ELEM( *object_points, float, i, 1) ;
    CV_MAT_ELEM( *object_points2, float, i, 2) = CV_MAT_ELEM( *object_points, float, i, 2) ;
}
for(int i=0; i<successes; ++i){ //These are all the same number
    CV_MAT_ELEM( *point_counts2, int, i, 0) = CV_MAT_ELEM( *point_counts, int, i, 0);
}
cvReleaseMat(&object_points);
cvReleaseMat(&image_points);
cvReleaseMat(&point_counts);
CV_MAT_ELEM( *intrinsic_matrix, float, 0, 0 ) = 1.0f;
CV_MAT_ELEM( *intrinsic_matrix, float, 1, 1 ) = 1.0f;
cvCalibrateCamera2(object_points2, image_points2, point_counts2, cvGetSize( image ),
intrinsic_matrix, distortion_coeffs, NULL, NULL,0);
cvFindExtrinsicCameraParams2(object_points2, image_points2, intrinsic_matrix,
distortion_coeffs, rot_vec, trans_vec, 0);

cvSave("Intrinsics.xml",intrinsic_matrix);
cvSave("Distortion.xml",distortion_coeffs);
cvSave("VectorRotacion.xml",rot_vec);
cvSave("VectorTrans.xml",trans_vec);
CvMat *intrinsic = (CvMat*)cvLoad("Intrinsics.xml");
CvMat *distortion = (CvMat*)cvLoad("Distortion.xml");

IplImage* mapx = cvCreateImage( cvGetSize(image), IPL_DEPTH_32F, 1 );
IplImage* mapy = cvCreateImage( cvGetSize(image), IPL_DEPTH_32F, 1 );
cvInitUndistortMap( intrinsic, distortion, mapx, mapy);

cvNamedWindow( "Undistort" );
while(image) {
    IplImage *t = cvCloneImage(image);
    cvShowImage( "Calibration", image );
    cvRemap( t, image, mapx, mapy );
    cvReleaseImage(&t);
    cvShowImage("Undistort", image);
    int c = cvWaitKey(15);
    if(c == 'p') {
        c = 0;
        while(c != 'p' && c != 27) {
            c = cvWaitKey(250);
        }
    }
}

```